An Experimental Performance Evaluation of Innovations Sequence-based INS Monitor against GNSS Spoofing

Martinez Jimenez, Pau and Tanil, Çağatay

Navigation Laboratory, Illinois Institute of Technology

ETSEIB, Polytechnic University of Catalonia

Aug 2018

Contents

List of Figures

1 Introduction

The human being is known for their thirst of knowledge, seeking answers to unanswered problems and figuring out better solutions for problems that have already been solved. This powerful sense of self improvement has brought civilization to invent agriculture, livestock, construction and transportation methods, among others.

With each new improvement, new tools were devised and new necessities arose that needed solution. When designing new methods of transportation, for example, at the beginning the use of rudimentary maps was the only option for navigation. With the help of some basic tools and knowledge of the constellations, one could find its approximate latitude fairly easily. The astrolabe, a tool devised for exactly this purpose dates the ancient Greeks.

Longitude, however, was an unsolved problem until the 18th Century, almost 2000 years later. Until then, it was common for sailors to navigate to the destiny's latitude and from there, sail East or West until reaching destination.It wasn't until the maritime chronometer became available that the possibility of knowing your position around Earth became possible.

But how come a clock can be used to estimate longitude? Knowing the time of a given location allows you to reproduce the state of the sky in that location, In particular, the position of the sun in the sky. The Marine chronometer allowed the ship's captain to keep track precisely of the time at the departure harbor, therefore knowing the sun's position. Comparing that position to the position of the sun seen by the ship allows to measure the distance between the two locations and therefore the latitude. This was the first precise way to measure the longitude for vehicles like ships, where the use of pendulum clocks or similar is impossible.

All navigation methods stayed practically the same until the 20th Century, 200 years later, when the localization problem was improved substantially once again. The installation of the GPS system around Earth became a reality, providing extremely accurate estimations of position and velocity, cheaply and effortlessly to a wide range of users around the world.

The GPS system is in essence, very similar to the Marine Chronometer method used by sailors. Instead of the sun, it is based in the localization of several satellites in orbit, with known positions, acting as a constellation that everyone can reference from.

In order to know your position, you only need to know the distance to the satellites in view. This is achieved by measuring the time of travel of a radio message sent by the space vehicles. Since radio waves travel at the speed of light, computing the distance is fairly simple, and with a few trigonometric calculations, the position of the user is found quickly.

This orbital lighthouses have revolutionized the way we live substantially. Nowadays, even our phones can quickly estimate our position to a few meters easily, contributing to a whole new set of applications, from the use in autonomous navigation to the study of tectonic plates or even space weather.

1.1 Navigation Systems

In this report we will focus the attention to the use of GPS for navigation purposes. In navigation, the common problem is to solve for the precise location of a given vehicle. The position is critical in application like piloting, path planning or autopilot, in the latter, the vehicle operates autonomously without pilot intervention, relying only on its sensors and thus the requirements precision and robustness are very strict.

It is common however to not only require the knowledge of position. Usually, the demand for the whole state of the vehicle is to be estimated. This includes not only position but also velocity and, in vehicles like planes and ships, attitude (yaw, pitch and roll). In cars for example, position, velocity and heading is usually sufficient. This application dependency requires the tailoring of the particular algorithm to a precise application for improved accuracy over conventional estimation methods.

For common ground applications, where the pilot is constantly on the wheel, the estimation is achieved fairly easily, the pilot itself is the sensing unit in charge of measuring the car's position and direction. the velocity is measured through a tachometer and displayed in the driver's dashboard for their more precise knowledge. This is the most basic navigation system but is also really effective.

In aerial and marine application however, where the background is uniform and repetitive, it is not trivial for a pilot to know their position without any reference point to estimate distances from. It's in this environment when the use of more advanced sensors for navigation is not only desired but necessary.

In the present days, there is a massive reliance on GNSS (GPS and other constellations) as an extra sensors for navigation. They allow for the computation of the position of the receiver cheaply and precisely, with a sensor of usually small footprint, ideal for applications where weight is critical.

A GPS receiver installed unaided is a simple and effective way to obtain a fairly precise position cheaply. In that sense, the invention of GPS has posed a dependence in satellite navigation that some malicious subjects are able to take advantage from.

1.2 Dangers

A plane flying near an airport, beginning its approach for landing is being piloted with the aid of GPS with ground augmentation systems. This is the use of a reference ground station to measure precisely biases and drifts in GPS signal.

This setup is known for achieving sub meter precision in their estimations. With this system, the pilot is able to know the height of the plane precisely, to perform the approach safely.

However, when approaching the landing track, the navigation system tells the pilot to descend slightly to correct their approach trajectory.

If that correction was caused by a malicious subject hijacking the GPS signals, this could pose a massive security threat to everyone involved, the airport, the passengers and the pilot.

Whenever a system is designed and it's failure could generate a clear danger to someone, it is necessary to develop countermeasures and safety features to protect against it.

In the case of GPS, a malicious subject could broadcast a GPS signal from a portable ground station more powerful than that of the satellites. In that case the GPS receiver would interpret that signal as genuine and bias the estimation of the position of the victim vehicle.

This kind of fault is called Signal Spoofing, and may be used for creating dangerous situations. For autonomous vehicles, a spoofed signal could be easily used to steal the vehicle by forcing it to follow a fake trajectory.

1.3 Spoofing

GPS spoofing is a serious threat with real implications. As such, it needs practical solutions, easy to implement so that nobody has an excuse to not use them.

Even though spoofing seems like a complex process, it can be achieved easier than expected. With the use of a GPS simulator, commonly used in laboratories around the world to test hardware and algorithms, it is possible to generate the same signals that a genuine satellite would generate and broadcast them to nearby devices.

This form of spoofing would classify as open loop spoofing, where the spoofed does not measure the position of the victim vehicle. If they use devices like laser ranging or radar or similar and feed that information to generate a spoofed signal that adapts to what the victim is doing, the spoofing is classified as a closed loop spoofing. As expected, the latter is harder to detect and thus more dangerous.

In this report we will define the Worst Case Fault, a particular spoofing signal that maximizes the risk for the vehicle while also minimizing the detection capability of the detector proposed.

The detector will be examined against this particular fault along with different more common types of fault to evaluate it's performance.

2 Objectives

The main objective of this research is to develop an algorithm capable of detecting GPS spoofing and evaluating its performance against different kind of faults.

This will be done with the use of real data measured in site with off-the-shelf sensors and components to prove it's applicability in a real navigation environment.

The faults tested will include different ramp faults with different accelerations profiles as well as the worst case fault for this particular algorithm.

Some robustness tests will be conducted to test the performance to the different spoofing scenarios including early and late monitoring.

This will be achieved incrementally through the development of basic sub-algorithms that can be easily tested for errors and inconsistencies. This sub-algorithms will then be incorporated together in the final detection algorithm and its performance will be tested.

This report also aims to give a comprehensive review of the basic working principles that this algorithm is based on. This will include stochastic processes, GPS and INS, estimation theory and Kalman filtering.

Then the inner workings of the algorithm will be explained as well as the specific parameters used and recommendations on how to design for accurate performance.

Lastly, the monitor's performance will be tested against several faults and a final evaluation will be extracted. This evaluation will include the performance against early and late attacks to different fault profiles to prove the algorithm's potential.

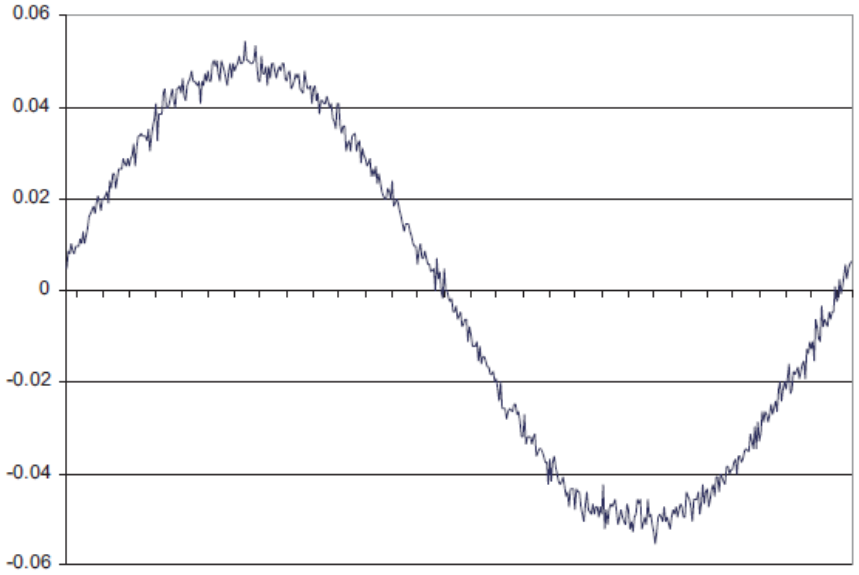


Figure 1: Oscilloscope example signal

3 Background

The background section will focus primarily on all the necessary skills and theory needed to follow the subsequent chapters based on the research needed to develop such algorithm.

Some background in controls, statistics and state space modeling is assumed and this report will work over that baseline to explain the necessary terms.

3.1 Stochastic Processes

Processes and signals when modeled conventionally in physics, mechanics or control are usually assumed to be smooth and continuously varying, an ideal signal. This signals exhibit well defined properties of its integrals and derivatives and are very useful for studying the behaviour of systems from a theoretical point of view.

However, when the modeling is done for implementation into a real world device or application, signals differ from their theoretical counterparts. Signals are different when measured, even the same signal measured through different devices or at different times will not look exactly the same.

This variability may be due to unmodeled effects, like electromagnetic radiation, or thermal effects, even ambient noise or pressure fluctuations. Reasons may vary depending on the specific application but the effect is the same, noisy signals.

These signals, when observed through an oscilloscope or a similar measuring device, fluctuate over time, not only as in the ideal signal but in other unpredictable ways.

A clear example of this effect is any electrical alternating signal (Figure 1). When measured, the obvious alternating signal is present but over that signal, a small noise can usually be observed that perturbates the signal locally around its true value. This is a common case of stochastic process, one of the most common, usually modeled as an additive white noise.

Usually, this unmodeled effects are small in magnitude and are far too complex to model with precision. To account for them, the theory of stochastic processes was proposed, the first step towards one of the most used filters to be devised, the Kalman filter.

There exist different types of stochastic processes, each with its own properties and characteristics. Choosing the right model to represent the uncertainty in the signal is left to the engineer in charge and will depend on the specific behaviour observed from the perturbations of the signal.

Stochastic processes are by definition impossible to model with analytic expressions, because of their chaotic nature. It is possible however, to measure their statistical properties, with which is possible to design a system and account for the variability the best way possible.

The simplest stochastic model and the base for several more complex models is the Gaussian white noise.

3.1.1 Gaussian white noise

A series of random values distributed over time that follows a Gaussian distribution is a common perturbation in signals. An example of this would be the random noise that a constant signal may have when measured with an oscilloscope.

If the Power spectral density (PSD) of said noise is constant over all frequencies, it is regarded as white, forming a Gaussian White noise when combining both characteristics.

$$PSD_w(\omega) = \int_{-\infty}^{+\infty} corr(w(t_1), w(t_2)) e^{-j\omega(t_2-t_1)} d(t_2 - t_1) = \sigma_w^2, constant \quad (1)$$

Having a constant PSD indicates that all frequencies of noise are equally probable, from infinitely slow to infinitely fast, which can be demonstrated to have infinite power. Infinite power signals are unfeasible but its ease of modeling however, makes it a really useful tool for generating other kind of noises, even if they are band-limited.

Approximating a real noise by a white noise is not limiting as long as the band-with of the real "white" noise is much larger than the band-with of interest of the particular application.

The vast majority of other interesting stochastic processes arise when a white noise is the input to a Linear System of equations. This transformations are

well known and easy to model in terms of Laplace transforms which make them ideal candidates to be used in a control environment.

3.2 Linear Systems with random noise inputs

Estimation systems often use Linear Systems as the tool of choice for modeling their inputs and outputs so it is only natural for us to model the stochastic processes as Linear Systems as well so the integration between both subjects is seamless.

Given a stochastic Linear system of the form:

$$\dot{x} = Fx + Gw \tag{2}$$

$$y = Hx + v \tag{3}$$

Where F, G and H are matrices, x is the state vector and w and v are the driving white noises.

This is the general form of an stochastic process written as a Linear System. Different values for the matrices in the model will define the behaviour of the system and its properties. Some examples of useful systems are:

3.2.1 Random Walk

If a random white noise is simply integrated:

$$y = \int_0^t w(\tau)d\tau \tag{4}$$

Or alternatively written in matrix form:

$$\dot{x} = \begin{bmatrix}0\end{bmatrix} x + \begin{bmatrix}1\end{bmatrix} w \tag{5}$$

$$y = \begin{bmatrix}1\end{bmatrix} x \tag{6}$$

This simple operation on the Gaussian White noise changes its behaviour drastically. Computing the variance of the output y through the Lyapunov Equation, it can be shown [1] :

$$var(y(t)) = \sigma_w^2 t \tag{7}$$

The variance of the random walk increases linearly with time. This effect has drastic implications, whenever a sensor signal is integrated, no matter how small the noise is, the precision of the integrated quantity will deteriorate over time.

Moreover, with every integration of the same signal, comes an increase in the speed of deterioration. In the case of navigation, is common to see a signal being integrated twice or three times, and as expected, without any extra help, the estimation of the position would be impossible.

This fact is one of the reasons justifying the existence of estimation theory and the search for ways to mitigate these effects. Several ways have been conceived, including the addition of observers, redundant sensors and the Kalman filter itself.

3.2.2 First Order Gauss-Markov Process

When a random noise is fed to a low pass filter, the resultant stochastic process is a Gauss-Markov of a certain order. The order of the process is determined by the degree of the filter used. A first order Gauss-Markov process is therefore a random white noise filtered by a first order low-pass filter, and the resulting system follows the dynamics:

$$\dot{x} = \frac{-1}{\tau}x + w \tag{8}$$

Where τ is the process' time constant or *correlation time*.

Applying the Lyapunov equation to solve for the evolution of the variance of the output we obtain:

$$var(x(t)) = \frac{\sigma_w^2 \tau}{2} + \left(var(x(0)) - \frac{\sigma_w^2 \tau}{2} \right) e^{-2\frac{t}{\tau}} \tag{9}$$

Which can be seen to have a finite steady state variance of $\frac{\sigma_w^2 \tau}{2}$ often used to estimate the correlation time.

In order to apply these continuous time stochastic processes one extra step is needed, the conversion to discrete domain. The conversion is done in the same manner as in control systems, through the use of zero order holds at a sampling rate determined by the sensor or measuring device.

Theses are the necessary models to describe the IMU and GPS sensors used in this research. This sensors can be classified as having good precision which simplifies the need for more complex stochastic processes.

One extra stochastic process will be used and will be explained in detail in Section 3.4.4.

3.3 IMU

Two main sensors will be used for localization purposes, an IMU and a GPS. Both will be explained in detail in the following sections.

An Inertial Measurement Unit (IMU) is a device capable of measuring the inertial forces that may be applied to its case. This forces contain information about the particular movement the IMU is following and therefore, can be used to extract several kinematic properties.

IMU usually include 2 basic sensors, an accelerometer and a gyroscope but they might be equipped with more. Typical IMU sensors usually include inclinometers and magnetometers, to precisely determine orientation in 3D space.

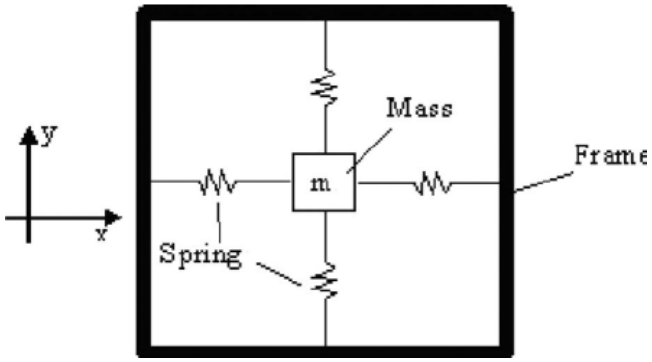


Figure 2: Basic model of an accelerometer

- The accelerometer, as the name implies,will measure linear acceleration in 3 perpendicular axis but as will be reviewed shortly, it will also measure the gravity vector which will corrupt the measurements.
- The gyroscope is a device that measure angular speed, usually in the same 3 axis as the accelerometer, which can be used to find the attitude of the vehicle through several trigonometric transformations.

3.3.1 Accelerometer

The accelerometer can be easily modeled as a mass suspended inside a case by springs (Figure 2).

In this setup the objective is to write a model in terms of the force applied on the springs, which we are able to measure electrically.

Applying the Newton's Second Law to the mass we obtain:

$$m\vec{a} = \vec{F_x} - c\vec{v} + m\vec{g} \tag{10}$$

Where $\vec{a}$ is the mass' acceleration, $\vec{F_x}$ is the spring force, c is the spring damping coefficient and g is the gravity field vector.

Since the actual implementation of this system is in form of a MEMS, the damping coefficient is small enough to be disregarded in the model. Making this assumption and solving for the specific spring force it can be seen:

$$\vec{f_x} = \frac{\vec{F_x}}{m} = \vec{a} - \vec{g} \tag{11}$$

The specific spring force $\vec{f_x}$ is measured in m/s^2 and is directly proportional to acceleration, making it the ideal candidate for the sensor output.

The actual implementation of such a sensor usually involves three different spring masses actuated in each separate axis, with micro capacitors to allow the measurement of mass' displacement and spring force. It can be demonstrated, however that the final model behaves similarly.

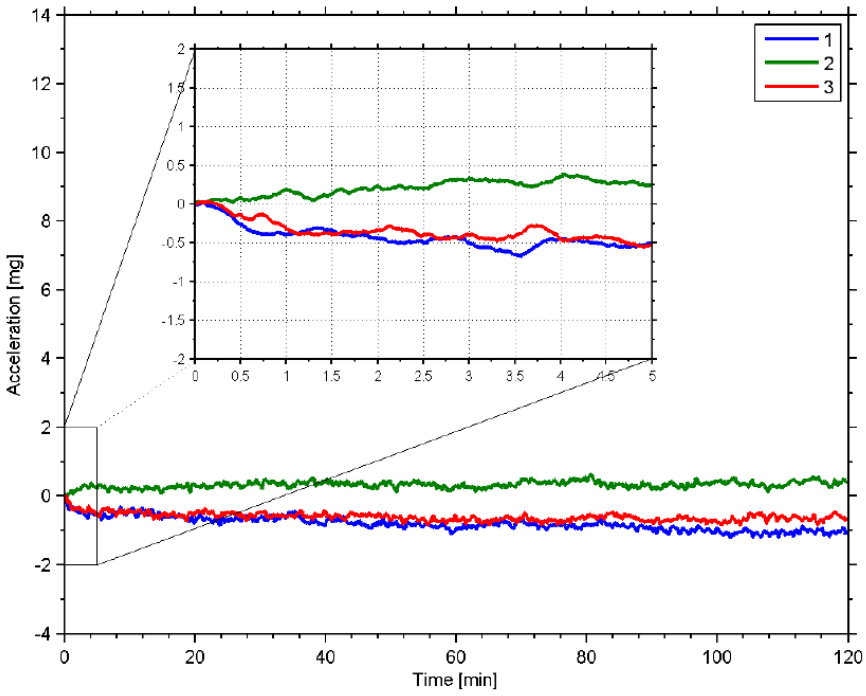


Figure 3: Typical normalized initial bias drift from STIM300 data-sheet

To account for all the unmodeled effects, such as thermo-electric and electro-mechanical perturbations as well as electromagnetic radiation and thermal fluctuations, we will use stochastic processes to model the uncertainty in the system.

$$\vec{u} = \vec{a} - \vec{g} + \vec{b_a} + \nu_a \tag{12}$$

The model proposed is a combination of a slow varying First Order Gauss-Markov process $\vec{b_a}$ added to a Gaussian White noise ν_a. The superposition of this processes aims to model the fast noise of any electrical signal plus the slow time varying bias drift that can be observed in this class of sensors (Figure 3)

As can be seen in the figure, the output value varies slowly until it reaches a steady state value with a constant covariance. This properties closely resemble the First Order Gauss-Markov process. To model the the variation over the mean value of each run, the model is complemented with a Gaussian White noise of adequate variance.

3.3.2 Gyroscope

The same procedure can be followed to obtain the model for the gyroscope sensor. However this sensors are based in oscillatory effects and Newton's Second

Law will not be of particular use in this case.

Instead, it is assumed that the output of of the gyro has been calibrated to output a value directly proportional to the angular speed of the case. The theoretical model:

$$\vec{w} = \vec{\omega} \tag{13}$$

The output however, is also corrupted by continuously varying biases and noise, in a similar fashion to the accelerometer. With that in mind, the actual process model for the gyroscope expands to:

$$\vec{w} = \vec{\omega} + \vec{b_w} + \vec{\nu_w} \tag{14}$$

The sensor output is corrupted by a slowly changing bias $\vec{b_w}$ and a white noise ν_w, adding up to a similar model to the accelerometer.

Even though they have a similar representation, the inner workings of a gyroscope are mostly based in the effects of the Coriolis acceleration to a vibrating lever arm. when its rotated, it induces a secondary oscillation to a sensing arm capable of measuring the displacement, which is proportional to the rotating speed.

3.4 GPS

GPS is a tool that has been fully operational for civil use since the 90s and as expected, the literature on the topic is extensive. The vast amount of research on the topic has brought unexpected uses for the system, like space weather monitoring or even the study of tectonic plates.

In this report however we will focus on GPS for positioning purposes, in particular, standalone systems with single frequency measurements. The aim of this section is to give a comprehensive review on the basics of GPS from a control and estimation point of view to be used later in subsequent chapters.

A GPS receiver is expected to perform 2 basic tasks, measuring the time of travel of the signal sent by the GPS satellites and then, computing an estimate for the position. This two basic operation can be subdivided in several other sub tasks, reaching from low level analog signal processing to a high level algorithm computation.

For navigation purposes, the data needed is the output of the first section of operations. Assuming that the receiver computes all the times of travel, and gives the necessary data to compute the satellite position, how is this information used to estimate the position of the receiver?

First, looking at the time of travel information by itself.

A receiver is capable of measuring 2 types of signals from a given satellite, pseudoranges and carrier phase measurements. These measurements come from 2 different sources but encode similar information. Pseudoranges is the message sent by the satellite itself, communicating the time at which the signal was sent. The carrier phase however, is a very accurate measurement of the phase of the incoming carrier radio wave.

With the measurement of the time of emission of the signal and comparing it to the time of reception, the distance to the satellite can be estimated. But since pseudoranges, use a digital signal to measure the time of travel, the error is quantized and thus its precision is low comparatively. This signal however, is easy to measure and to process, which aids in the speed for computing more accurate estimates.

The receiver can also track the phase of the incoming carrier signal, this information can be used to estimate the position of the receiver to a cm level but it has a significant drawback. Since the signal wave is sinusoidal, the position is well know but it has periodic solutions, one each wavelength (around $20cm$).

This ambiguity prevents the knowledge of the absolute position but the information provided can be useful if the satellite is monitored continuously maintaining the ambiguity constant. Carrier measurements can be used to smooth the code or be used along the as extra measurements that will reduce the uncertainty in the position.

A good mathematical model will include these effects as well as several clock delays and some environmental perturbations. A common measurement model for GPS signals:

$$\begin{aligned} \rho &= l_i^{(k)} + c\tau_i - c\tau^{(k)} + I^{(k)} + T^{(k)} + m_\rho^{(k)} + \nu_\rho^{(k)} \\ \lambda\Phi &= l_i^{(k)} + c\tau_i - c\tau^{(k)} + I^{(k)} + T^{(k)} + m_{\lambda\Phi}^{(k)} + \nu_{\lambda\Phi}^{(k)} + \lambda N^{(k)} \end{aligned} \tag{15}$$

This set of equations represents both code ρ and carrier $\lambda\Phi$, where $l_i^{(k)}$ is the geometric distance between the satellite at the time of signal transmission and the user at the time of reception. The perturbations can subdivided in 3 sections, clock delays, atmospheric perturbations, and stochastic processes. In this formulation, the subindex i represents a user parameter and the superindex (k) represents a satellite.

The range estimation is based in the measure of the time of travel of a radio signal divided by the speed of light. It is obvious to see that if the clocks in both receiver τ_i and satellite $\tau^{(k)}$ are not aligned to the GPS reference time, there will be a measurement error. Satellite clocks are usually cesium or rubidium clocks. This clocks are very stable but drift over time. To account for this delay, the delay is measured at a ground station and fit to a polynomial curve. The coefficients of this parameters are sent to the satellites to be broadcasted down to all receivers in the navigation message.

Since the satellite clock delay can be estimated empirically, is fairly straightforward to compensate it. User clock bias on the other hand, cannot be directly measured. Receivers are manufactured to be affordable and their clocks are usually based in quartz oscillators, much less stable than the satellite's. This and the fact that there is not a reference to compare against, makes the user clock bias a more complex estimation than previously expected. The method for accounting for the clock bias will be discussed in the following section.

3.4.1 Atmospheric errors

GPS signals are emitted by the satellite in space and travel down to earth crossing all the atmosphere in the process. It is well known that light travel at lower speeds in mediums different from vacuum. Travelling from a satellite to a receiver only around 5% of the journey can be considered as the vacuum of space [2].

A photon emitted from a satellite must travel two sections of the atmosphere that affect travel time significantly. This sections are the Ionosphere and the Troposphere. Signals travelling through these mediums are altered and the travel time varies significantly to the point that needs to be accounted for navigation purposes.

3.4.2 Ionosphere

The Ionosphere is the upper region of the atmosphere, exposed to the activity of the sun directly where most of the ultraviolet radiation is stopped. When the radiation from the sun reaches the molecules of the Ionosphere (Mostly H_2, He, N_2 and O_2) the high energy photons excite their electrons and detach from the nucleus. This generates free electrons and positive ions, which cause the GPS signals to altered.

This alterations are proportional to the amount of free electrons along the path of the signal. The measure of the amount of electrons in the path of the signal is measured as the Total Electron Count (TEC). This quantity measures the density of electrons in a tube of $1m^2$ of section in the path of the signal. The path of the signal will change as the satellite travels through the sky which affects the TEC, being minimum when the satellite is at the zenith and increases as the satellite sets. Since the amount of electrons depend on the radiation of the sun, the TEC will strongly depend on the time of the day. The sun's activity on a particular region of the Ionosphere is minimal at night but increases as the sun raises and is maximum around 2pm. Other alterations in the sun activity like solar storms also affect the Ionosphere strongly, most significantly at the magnetic equator and the poles.

For GPS signals in the L band, the Ionosphere can be classified as $dispersive$, which means the refractive index depends on the frequency of the signal. The most significant effect of this factor is that carrier and code will be altered differently, causing what is known as $code - carrier\ divergence$.

An amplitude modulated signal is composed of 2 signals, the message signal with a significantly lower frequency that the carrier signal. In a dispersive medium like the Ionosphere, this two signals will be shifted differently. In particular for code and carrier in the GPS L band, This effect delays the modulated signal (code) but advances the carrier. This divergence is equal in magnitude but opposite in sign and the compensation is not trivial.

3 common approaches to compensating Ionospheric delays:

Some receivers are equipped with double frequency measurements, which allow them to measure the 2 GPS signals broadcasted by the Space Vehicles thus

providing enough redundancy to solve for the filtered signal. This signal however will be noisier than the uncorrected signal by a factor of 3 approximately.

Another approach would be to use a Ground Based Augmentation System (GBAS) to compensate for all combined delays of the signal. Planes approaching an airport usually have a GBAS at their disposal. This is a reference station with a known position that allows to measure the state of the sky around the antenna. This allows the plane to have a *cm* level position when executing critical maneuvers like approaches and landings.

GBAS however require an expensive and well calibrated setup and their range is limited.

The last approach commonly used in standalone receivers is the use of Ionosphere Delay models that coarsely represent the delays and compensate for them. Nowadays, there isn't a model that precisely predicts the Ionosphere behaviour so the precision of the model is limited. This is however an easy and affordable approach when extreme precision is not necessary. The most common model is the Klobuchar model, whose parameters are broadcasted in the navigation message and updated frequently.

Choosing an approach will depend on the particular application and the requirements of precision. For this demonstration, where the objective is to show a spoofing detection algorithm, we will choose to use simulated data to which the Ionospheric delays can be switched off to not distract from the bigger picture.

3.4.3 Troposphere

Another significant source of error in GPS transit time is the effect of the Troposphere. The troposphere is the lower part of the Atmosphere where living beings reside. This part of the atmosphere is mainly composed of dry gases and water vapor.

The Troposphere, contrary to the Ionosphere, is non dispersive, with a refractive index slightly higher than 1 ($n \simeq 1.003$) and it delays all signals equally, as if satellites where found from 2.5 to 25m further than their actual position.

An equal delay on all signals cannot be compensated through algebraic operations on the signals and it needs to be compensated via models or with reference stations. Several models have been proposed that use meteorological data to estimate the Tropospheric delay but is not possible to compensate simply like the Ionospheric delay with dual frequency measurements.

As mentioned before, in this research the use of simulated data is used and the Tropospheric effects on the signal can be ignored.

3.4.4 GPS Stochastic Delays

The last terms in GPS model includes 3 different terms, integer ambiguity, multipath and noise.

Integer ambiguity, which has been mentioned previously, is the uncertainty on the solution of the carrier phase measurements. This ambiguity can be

modeled with an stochastic model to account for its uncertainty. The ambiguity is a constant offset in the carrier measurement that does not change over time as long as the satellites are tracked continuously. If there is a loss of lock, the value will change but will remain constant afterwards. To model this behaviour we will make use of the Random constant model.

A Random constant is a value that remains constant over time but it's initialized with an unknown value but known distribution. The system does not have dynamic properties but it introduces uncertainty to the model. This model perfectly describes the behaviour of the integer ambiguity and will be used to model the carrier phase.

The second kind of perturbation is multipath. This is the effects of the reflection of the radio signal on reflective surfaces, walls, the ground, or even mountains. This reflection perturbates the radio signal, adding a variation that is related to the geography of the localization of the receiver.

Multipath is an issue that can be attenuated with good design but it is impossible to eliminate completely. Adding an antenna with a gain pattern that rejects ground reflections will reduce that effect. Some other solutions include adding the antenna to the roof of the vehicle, like a plane, so all reflected signals from below are shielded from the receiver by the metallic fuselage of the plane.

But given a fixed design, multipath will still contaminate the signal depending on the environment of navigation, which results in a very intensive modeling effort if all the buildings and geometry are to be reproduced. When examined from a stochastic point of view, it commonly presents the properties of a First order Gauss-Markov process, in the same way as the IMU sensor, it will have however, a shorter correlation time, whose value generally depends on the particular geometry.

The last stochastic process that affects the GPS signal in a significant way is a white noise around all the signals due to the signal reception and processing, usually referred to a Thermal Noise, it is not only caused by the random photons generated hot bodies, it is an aggregate of all similar noises that affect the system that does not have an obvious structure.

3.4.5 Simplified model

With all the deterministic delays being accounted for, the model can be reduced to:

$$\begin{aligned} \rho_{IF} &= l_i^{(k)} + c\tau_i + m_\rho^{(k)} + \nu_\rho^{(k)} \\ \lambda\Phi_{IF} &= l_i^{(k)} + c\tau_i + m_{\lambda\Phi}^{(k)} + \nu_{\lambda\Phi}^{(k)} + \lambda N^{(k)} \end{aligned} \tag{16}$$

This model only accounts for unknown drifts and stochastic processes and further simplification is complex.

It is however possible to account for the user clock delay with a small trade of. If the number of measurements can be reduced by 1, it is possible to eliminate clock bias from the measurement model.

Choosing a satellite (r) as a reference satellite, allows to measure differences between ranges, which are not affected by the user clock bias:

$$\begin{aligned}\rho_{IF}^{(k)} - \rho_{IF}^{(r)} &= l_i^{(k)} - l_i^{(r)} + m_\rho^{(k)} - m_\rho^{(r)} + \nu_\rho^{(k)} - \nu_\rho^{(r)} \\ \lambda\Phi_{IF}^{(k)} - \lambda\Phi_{IF}^{(r)} &= l_i^{(k)} - l_i^{(r)} + m_{\lambda\Phi}^{(k)} - m_{\lambda\Phi}^{(r)} + \nu_{\lambda\Phi}^{(k)} - \nu_{\lambda\Phi}^{(r)} + \lambda N^{(k)} - \lambda N^{(r)}\end{aligned} \tag{17}$$

This operation comes with the lost of a measurement equation, and an increase in noise in the measurement equation. Note that variances are always additive and taking the difference of 2 stochastic processes will add the variance of each one. This is the reason why this operation is usually performed with the satellite with the highest elevation in the sky. The highest satellite has the smallest multipath due to the fact that the signal has less possibility of reflecting against surfaces.

3.4.6 Satellite position

The computation of position is impossible without a reference to compute it from. In the GPS system, this reference are the satellites themselves, acting as radio signal lighthouses with known positions.

The position of a satellite is broadcasted in what is referred to as the *Ephemeris*. The *Ephemeris* is a set of 16 constants that are broadcasted along with the measurements. The constants are used in a mathematical model to estimate the position of the satellite at any time of need. This allows the user to compute the satellite position whenever is possible and not needing to wait for the satellite to broadcast the position with each measurement.

The Ephemeris is updated every few hours to ensure the precision of the measurements to a certain specification.

The parameters of the Ephemeris follow a Keplerian model with time varying parameters. This means the orbits of the satellites are measured as general ellipses whose parameters are allowed to change slightly over time. This model, despite not being the most precise, it was chosen as a compromise between amount of parameters transmitted in the navigation message and precision. In the years when GPS was designed, computers and electronics had limited capabilities and the standard has been kept since then.

The existence of GPS has allowed for precise positioning anywhere on the globe that is easily automated. A navigation system wouldn't be possible without the use of GPS for tracking the position of the vehicle directly without any signal integration. GPS however, measures the position with a significantly lower frequency of around 1Hz compared to the several hundred Hz of inertial sensors, which complicates its use for control purposes or to estimate other states, like velocities.

Proceeding with this measurement model forward, the navigation system will be developed to use this information to estimate the position of the receiver with the best precision possible.

3.5 Observers

Implementing a navigation system is possible through the use of several algorithms, like least squares estimators or the Kalman filter that we will be using. A Kalman filter is nothing more than a special Observer model that is capable of estimating the states of the system precisely. This estimation is done possible through a feedback loop, which corrects the estimations of the primary sensors with the information of aiding sensors.

The mathematical description of the observers is the following:

An estimation system, defined by the state space equations below, is used to accept a sensor input u and compute the solution of the state x. This approach is widely used in control and robotics and can be extrapolated to the GPS use.

$$\dot{x} = Fx + G_u u \tag{18}$$

If in this configuration, if an extra measurement of an aiding sensor is available y, and this measurement can be modeled as:

$$y = Hx \tag{19}$$

Then it is possible to compare the value obtained by the sensor $\widetilde{y}$ to what we expect our system to measure y. If that difference is then added as a correction to the propagation through a gain matrix L, we are able to correct the estimation of the system with aiding sensors

$$\dot{x} = Fx + G_u u + L(\widetilde{y} - y) \tag{20}$$

It can be proved that as long as the eigenvalues of $F - LH$ are stable and the system has the mathematical condition of being observable, the system will converge to its true value even if the system has noise [1].

This is a remarkable result and is left to the engineer to choose their desired gain matrix L to make the convergence faster or slower.

3.6 Kalman Filter

Choosing a particular gain matrix L based on pole placement is an straightforward approach that is commonly used so the estimated states converge to the real value faster than the dynamics of the real system. But in general choosing the gain matrix can be done by selecting any criteria.

If that criteria is minimizing the covariance of the next state estimation, a general observer becomes a Kalman filter. This filter minimizes the covariance matrix of the next state estimation by selecting a gain matrix calculated with information of all the signals' noise.

The derivation of the Kalman filter is based on the idea of minimizing the covariance of the state vector just after the observer's correction. Writing the correction with the following notation:

$$x_k^+ = x_k^- + L(\widetilde{x} - Hx_k^-) \tag{21}$$

Where x_k^- is the state estimation prior to being corrected and x_k^+ after the correction, The covariance of this expression under the Kalman filter assumptions [3] will be:

$$P_k^+ = (I - LH)P_k^-(I - LH)^t + LRL \tag{22}$$

Here P_k^+ and P_k^- are the covariance matrices of the a priory and post correction, I is the Identity matrix and R is the covariance matrix of the measurements.

The key to the Kalman filter derivation is to minimize the trace of the covariance of the previous expression to obtain:

$$L = P_k^- H^t (H P_k^- H^t + R)^{-1} \tag{23}$$

As can bee seen this expression uses noise data to compute the gain matrix of the filter which implies that it will change at each computation step, which makes the Kalman filter a time varying system.

The standard procedure for a Kalman filter implementation, which combines all the necessary operations to compute the necessary matrices is the following:

- **Initialization**: It is necessary to choose the best estimation possible for the initial state and its covariance x_0^- and P_0^-.
- **Measurement Update**: Then, when a measurement is available, the algorithm proceeds to update and correct the states estimations through the following set of equations:

$$L_k = P_k^- H^t (H P_k^- H^t + R)^{-1} \tag{24}$$

$$z_k = \widetilde{y} - y \tag{25}$$

$$x_k^+ = x_k^- + L z_k \tag{26}$$

$$P_k^+ = (I - LH)P_k^- \tag{27}$$

  This update will correct the value of the states to account for the new information coming from the measurement.

  Note: the difference between the measured $\widetilde{y}$ and the predicted y is also called the innovation γ_k

- **Time propagation** Next step is to compute the future state through the use of the model of the system, this can be done through any means available, from exact integration to numerically propagating the matrices. An example of propagation would be to use the discrete model equations as follows:

$$x_{k+1}^- = \Phi x_k^+ + \Gamma_u u \tag{28}$$

$$P_{k+1}^- = \Phi P_k^+ \Phi^t + Q_d \tag{29}$$

Here greek letters have been chosen to represent the discrete versions of the continuous time matrices F (Φ) and G_u (Γ_u). Qd is the covariance matrix of the primary sensor measurements and the driving noise of the aiding states.

This algorithm minimizes the noise in the state vector an therefore the state estimation is as precise as possible. It is important to note that since the convergence velocity cannot be set to any specific value, the convergence time will depend on the model chosen. The longer the system's time constant, slower the convergence.

This section concludes the necessary material for the implementation of the navigation system. The following sections will review the design and implementation of the particular navigation system used to monitor the spoofing attacks.

4 Implementation

With all the necessary knowledge for the implementation, this section aims to explain the design, inner workings and performance of the navigation system

The Navigation algorithm consists in a Kalman integration between GPS and INS to predict the position, velocity and attitude as well as all the aiding states of a vehicle. This system is based on the theoretical development of Professor Cagatay Tanil Thesis [4].

The system will use the single difference method to estimate GPS ranges, for a single frequency standalone receiver with pseudoranges and carrier phases. A tightly coupled integration to a the IMU mechanization equations will allow it to acquire estimations of all the states of the vehicle at a high rate, correcting for noise integration at each measurement epoch.

The mechanization equations will be discussed and later, the details on the particular implementation will be explained. Performance for different scenarios will follow and spoofing detection capabilities will be measured and quantified.

4.1 Mechanization equations

The mechanization equations is the set of kinematic equations that allow to compute the position of a vehicle in 3D space through the integration of acceleration and angular rates. This equations are used to compute the actual position of the vehicle from the accelerometer and gyroscope sensor outputs.

The mechanization equations will compute the position in North-East-Down (NED) frame, centered around a given initial position fixed in Earth frame. The algorithm accounts for 3 different frames. Navigation frame (N) is fixed to the ground following Earth's rotation. Earth frame (E) is an inertial frame of reference located on Earth's center but non-rotating. The last frame of relevance is the Body frame (B), fixed to the vehicle and the center of which we want to know the position. This frame has the axis aligned withe the axis of the vehicle and to the sensor's axis. For a visual representation, refer to Figure 4.

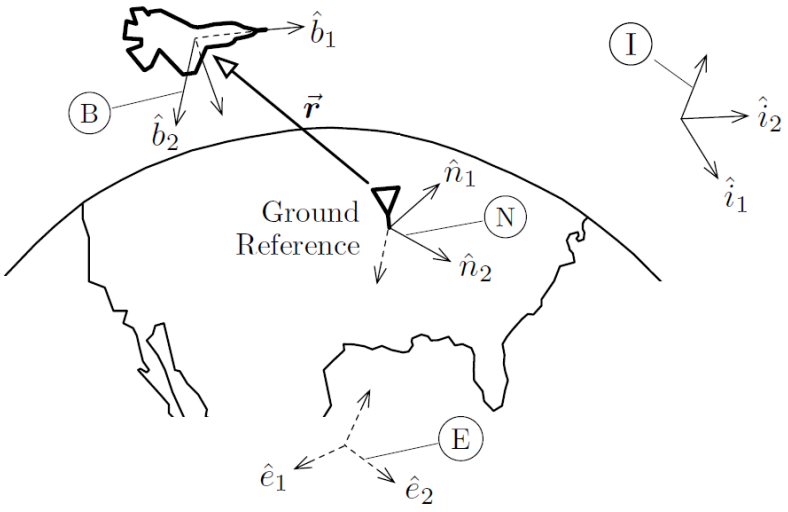


Figure 4: Schematic of the mechanization system

We define the position $\vec{r}$ as the position of the IMU in reference to a frame fixed to the ground (N). The first equation defines the velocity as the time derivative of $\vec{r}$ with respect to an inertial frame and written in the (N) frame of reference. To shorten notation we define $\frac{^I d(\cdot)}{dt} \triangleq \dot{(\cdot)}$.

$$\vec{r} = \vec{r}^{IMU}_{(N)} \tag{30}$$

$$\vec{v} = \frac{^I d\vec{r}}{dt} = \dot{\vec{r}} \tag{31}$$

The second mechanization equation, computes the variation of the velocity through the acceleration. However, as stated before, the acceleration is measured in the body frame (B) and thus a frame rotation is needed. Defining $^I R^B$ as the rotation matrix that transforms vectors in (B) to vectors in (N) we can write:

$$\dot{\vec{v}} = {}^N R^B \vec{a}_{(B)} \tag{32}$$

For this implementation we will make use of the North-East-Down frame definition (NED). As the name implies, frame (N) will have its first vector pointing North, the second pointing to the local East, and the third point down towards the Earth. For the body frame (B) a similar definition is followed. Its first axis will be pointing at whichever direction forward is, the third will be pointing down fixed to the vehicle and the second axis will be perpendicular to the other 2.

Note that if the vehicle is at the origin of the (N) frame, flat on the ground and pointing North, the axis of both frames coincide. This definition will allow for a simpler interpretation of the Euler angles.

In order to describe the attitude of the vehicle, and thus the ${}^{N}R^{B}$ matrix, several approaches can be used. Here it has been chosen the use of Euler angles description. Despite their singularities at certain orientations, their implementation is simple and their interpretation is straightforward.

Three consecutive Euler angles are needed to define the orientation of a rigid body in 3D space. We will use Yaw, Pitch and Roll as the rotations of choice. This approach is similar to the one used when studying aerial vehicles.

Starting with the axis of both frames coinciding, the first Euler rotation will be Yaw (ψ), defined as a rotation around the third axis of the (N) frame. ψ will rotate the vehicle and set it in a course in the 1-2 plane. The second Euler angle, Pitch θ is defined as the nose inclination of the vehicle with respect to the previous 1-2 plane. This angle defines the rate of climb or descent of the vehicle. The third and last rotation is Roll ϕ, defined as the rotation about the axis of the vehicle pointing forward.

With this 3 rotations defined, the Rotation matrix can be written as:

$$ {}^{N}R^{B} = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\ c(\theta)s(\psi) & cos(\phi)s(\psi) + s(\psi)s(\theta)s(\phi) & c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\psi) & c(\phi)c(\theta) \end{bmatrix} \tag{33} $$

The last kinematic equation involves the evolution of this Euler angles over time. We will write that evolution in terms of the angular velocity of the vehicle, because it can be measured through the IMU's gyroscope.

The last equation can be written from the fact that the sum of all three Euler rotations (After being expressed in the correct frames of reference) must be equal to the angular velocity of the body itself.

$$ \vec{\omega} = {}^{B}R^{N}\vec{\psi} + {}^{B}R^{A}\vec{\theta} + \vec{\phi} \tag{34} $$

Where the frame of reference (A) is the Body frame after being subject to the first and second rotations.

This expression can be simplified by substituting the rotation matrices by their expressions and computing the products. After some algebraic arrangements, the expression can be written as:

$$ \vec{\omega} = {}^{B}Q^{E}\dot{\vec{E}} \tag{35} $$

Where ${}^{B}Q^{E}$ is the matrix projection of the Euler angle vector into the body frame.

In this representation:

$$ \vec{E} \triangleq \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{36} $$

As can be seen, the equation relates the body rates as a function of the Euler angles but we are interested in the inverse. The inverse representation can be written as:

$$\dot{\vec{E}} = {}^{E}Q^{B}\vec{\omega} \tag{37}$$

$${}^{E}Q^{B} = [{}^{B}Q^{E}]^{-1} = \begin{bmatrix} 1 & sin(\phi) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \tag{38}$$

This third equation defines the complete set of the kinematic equations, which can be written in vector form as the following:

$$\begin{bmatrix} \dot{\vec{r}} \\ \dot{\vec{v}} \\ \dot{\vec{E}} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ {}^{N}R^{B}\vec{a}_{(B)} \\ {}^{E}Q^{B}\vec{\omega} \end{bmatrix} \tag{39}$$

If the acceleration and angular rates are substituted by the signals measured by the sensors, and the inertial forces of (N) with respect to (E) are computed, we obtain:

$$\begin{bmatrix} \dot{\vec{r}} \\ \dot{\vec{v}} \\ \dot{\vec{E}} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ {}^{N}R^{B}\vec{a}_{(B)} \\ {}^{E}Q^{B}\vec{\omega} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ {}^{N}R^{B}\vec{u} - \omega_{ie} \times (\omega_{ie} \times +\vec{r}) + \vec{g}_{earth} \\ {}^{E}Q^{B}(\vec{\omega} - \omega_{ie}) \end{bmatrix} \tag{40}$$

Where ω_{ie} is the earth rotation vector.

This model will be used to propagate the the IMU output to calculate variations in position, velocity, and attitude.

In order to apply the control theory to the previous equations, linearization is needed. Liniarizing will allow to write the model in terms of its F and G matrices and thus apply all the previous discussion for observers and the Kalman filter.

The first mechanization equation is linear by itself so no modification is needed. Writing it in terms of the state vector $x = [\delta\vec{r}, \delta\vec{v}, \delta\vec{E}]^t$ and the control vector as $u = [\delta f, \delta w]$, we obtain:

$$\dot{\delta\vec{x}} = [0_{3\times3}, I_{3\times3}, 0_{3\times3}]x \tag{41}$$

The second equation, which rotates the sensor output through the Euler angles rotation matrix and adds several terms, can be written in matrix form as follows:

$$\dot{\delta\vec{v}} = [0_{3\times3}, -2\omega_{ie_\times}, {}^{N}R^{B}f_\times]x + [{}^{N}R^{B}, 0_{3\times3}]u \tag{42}$$

Where the $\times$ operator indicates the the skew matrix operator of a vector.

Lastly, the third mechanization equation, due to it not having the convenient properties of rotation matrices, computing its derivative is troublesome in matrix notation. It can be written as follows:

$$\dot{\delta\vec{E}} = [0_{3\times3}, 0_{3\times3}, K]x + [0_{3\times3}, {}^{E}Q^{B}]u \tag{43}$$

Where K is a matrix of derivatives of the form:

$$K = \begin{bmatrix} \frac{\mathrm{d}\dot{\vec{w}}}{\mathrm{d}E} \end{bmatrix} \tag{44}$$

Grouping the equations:

$$\dot{x} = \begin{bmatrix} 0 & I & 0 \\ 0 & -2\omega_{ie_\times} & {}^N R^B f_\times \\ 0 & 0 & K \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ {}^N R^B & 0 \\ 0 & {}^E Q^B \end{bmatrix} u \tag{45}$$

Renaming the matrices as F and G_u we can write the equations as follows.

$$\dot{x} = Fx + G_u u \tag{46}$$

This system of equations will now be augmented with the equations belonging to the error model of the measurements so to include the biases and multipath states in the formulation. In this formulation the new state vector $x_a = [x, b_a, b_w, m_\rho, m_{\lambda\phi}, \lambda N]^T$. The augmented state matrix:

$$\dot{x}_a = \begin{bmatrix} F & -G_u & 0 \\ 0 & -1/\tau_b & 0 \\ 0 & 0 & -1/\tau_m \end{bmatrix} x_a + \begin{bmatrix} G_u \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} -G_u & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \nu \\ \omega_b \\ \omega_m \end{bmatrix} \tag{47}$$

This equation reflects all the necessary states that define the error signals of the measurement and the process noise. In this representation, ν, ω_b and ω_m are the white noise process of the IMU together with the driving noises of the bias and multipath states respectively.

Computing the discrete time equivalent of this system, we obtain:

$$x_{ak+1} = \Phi_a x_{ak} + \Gamma_u u_k + \bar{w}_k \tag{48}$$

The mechanization equations derived from this model ignore the noise terms since they are not measurable. The mechanization equations will be used to propagate the states over time in the time update phase of the Kalman filter.

$$\hat{x}_{ak+1} = \Phi_a \hat{x}_{ak} + G_{ua} u_k \tag{49}$$

4.1.1 Measurement model

Next step is to define the measurement model for the Kalman filter update. This includes how to compute the corrected pseudoranges as well as satellite position to define the H matrix of the Kalman update.

From equation (15) we know:

$$\begin{aligned} \rho^{(k)} &= l_i^{(k)} + c\tau_i - c\tau^{(k)} + I^{(k)} + T^{(k)} + m_\rho^{(k)} + \nu_\rho^{(k)} \\ \lambda\Phi^{(k)} &= l_i^{(k)} + c\tau_i - c\tau^{(k)} + I^{(k)} + T^{(k)} + m_{\lambda\Phi}^{(k)} + \nu_{\lambda\Phi}^{(k)} + \lambda N^{(k)} \end{aligned} \tag{15}$$

To compensate for the satellite clock bias term $\tau^{(k)}$, the IS-GPS-200 recommends the use of the correction polynomial broadcasted in the navigation message. This polynomial has the following form:

$$\Delta t_{sv1} = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_r \tag{50}$$

In this expression, the terms a_{f0} , a_{f1} , a_{f2} and t_{oc} are the parameters broadcasted in the navigation message, 3 polynomial coefficients and a time reference epoch respectively. This delay has to also account for relativistic effects due to the gravity field potential. The term Δt_r has the form:

$$\Delta t_r = \frac{-2\sqrt{GM}}{c^2} e^{\sqrt{A}} sin(E_k) \tag{51}$$

This correction term uses data from the satellite's orbit (Also broadcasted in the navigation message) as well a some universal constants. The $\sqrt{A}$, and E_k are orbit parameters while G, M, c are the Gravitational Constant, mass of the Earth and the Speed of light in a vacuum respectively.

This corrections would be sufficient if the measurements in our model were to be double frequency. Another correction is necessary when only a single frequency is considered.

$$\tau^{(k)} = \Delta t_{sv1} - T_{gd} \tag{52}$$

The T_{gd}, the group delay correction will reduce the error in pseudorange measurements in approximately 1 meter. As can be seen with this corrections, several depend on the orbit parameters and thus this equations will be coupled to the orbit estimation algorithm.

In order to compute the satellite's orbit and its position, the basic data needed are all of the Ephemeris parameters for that given space vehicle. The ephemeris contains 16 orbital parameters and allow for the use of the Keplerian model to estimate the position of the space vehicle at a given transmission time.

The algorithm for computation given a transmission time of t:

$$t_k = t - t_{oe} \tag{53}$$

$$M = M_0 + \left(\sqrt{\frac{GM}{A^3}} + dn \right) t_k \tag{54}$$

$$E_k = M + e \sin(E_k) \tag{55}$$

$$f_k = \tan^{-1} \left(\frac{\sqrt{1 - e^2} \sin(E_k)}{\cos(E_k) - e} \right) \tag{56}$$

$$\Omega = \Omega_0 + (\dot{\Omega} - \omega_{ie}) t_k - \omega_{ie} t_{oe} \tag{57}$$

$$\phi = 2(w + f_k) \tag{58}$$

$$w_k = w + f_k + C_{uc} \cos(\phi) + C_{us} \sin(\phi) \tag{59}$$

$$r_k = A(1 - e\cos(E_k)) + C_{rc}\cos(\phi) + C_{rs}\sin(\phi) \tag{60}$$

$$i_k = i_0 + \dot{i}t_k + C_{ic}\cos(\phi) + C_{is}\sin(\phi) \tag{61}$$

$$X_{sv} = \cos(w_k)r_k\cos(\Omega) - \sin(w_k)r_k\cos(i_k)\sin(\Omega) \tag{62}$$

$$Y_{sv} = \cos(w_k)r_k\sin(\Omega) + \sin(w_k)r_k\cos(i_k)\cos(\Omega) \tag{63}$$

$$Z_{sv} = \sin(w_k)r_k\sin(i_k) \tag{64}$$

With the SV position $[X_{sv}, Y_{sv}, Z_{sv}]$ computed, there's only one extra correction needed and that is to account for the rotation of earth during the time of travel of the radio signal. This last correction is coupled to the computation of the receiver clock bias. This is accounted through iteratively computing both terms until the solution converges.

$$\theta = \omega_{ie}\frac{\rho_k - \tau_i}{c} \tag{65}$$

$$X'_{sv} = X_{sv}\cos(\theta) + Y_{sv}\sin(\theta) \tag{66}$$

$$Y'_{sv} = -X_{sv}\sin(\theta) + Y_{sv}\cos(\theta) \tag{67}$$

For the application of this algorithm, the Ephemeris provides the following parameters (Figure 5), to which it needs to be added the constants G, M, c and the pseudoranges ρ_k

Computing the satellite position when a measurement is available is the first part of the measurement update. The second part includes the pseudorange correction and model. In this implementation, the method used will be the Single difference to a reference satellite outlined in section 3.4.5. The method accounts for the user clock bias solving for equations 17:

$$\begin{aligned} \rho_{IF}^{(k)} - \rho_{IF}^{(r)} &= l_i^{(k)} - l_i^{(r)} + m_\rho^{(k)} - m_\rho^{(r)} + \nu_\rho^{(k)} - \nu_\rho^{(r)} \\ \lambda\Phi_{IF}^{(k)} - \lambda\Phi_{IF}^{(r)} &= l_i^{(k)} - l_i^{(r)} + m_{\lambda\Phi}^{(k)} - m_{\lambda\Phi}^{(r)} + \nu_{\lambda\Phi}^{(k)} - \nu_{\lambda\Phi}^{(r)} + \lambda N^{(k)} - \lambda N^{(r)} \end{aligned} \tag{17}$$

The next step in the development of the measurement model is to linearize the measurement equations to obtain the measurement matrix H.

The nonlinear part of the measurement equations (17) is the actual geometric range $l_i^{(k)}$. This term is the euclidean distance between the receiver at reception time and the satellite at transmission time. The $\vec{r}_i = [x_i, y_i, z_i]$ are the receiver coordinates.

$$l_i^{(k)} = \sqrt{(X_{sv} - x_i)^2 + (Y_{sv} - y_i)^2 + (Z_{sv} - z_i)^2} \tag{68}$$

Using an initial point $\vec{r}_{i0} = [x_{i0}, y_{i0}, z_{i0}]$ as the Taylor expansion origin, the previous equation can be approximated by the following expression:

$$l_i^{(k)}(\vec{r}_i) \simeq l_i^{(k)}(\vec{r}_{i0}) - (\vec{e}_i^{(k)})^T \cdot \begin{bmatrix} (x_i - x_{i0}) \\ (y_i - y_{i0}) \\ (z_i - z_{i0}) \end{bmatrix} \tag{69}$$

Parameter	No. of Bits**	Scale Factor (LSB)	Effective Range***	Units
IODE	8			(see text)
C_{rs}	16*	2^{-5}		meters
Δn	16*	2^{-43}		semi-circles/sec
M_0	32*	2^{-31}		semi-circles
C_{uc}	16*	2^{-29}		radians
e	32	2^{-33}	0.03	dimensionless
C_{us}	16*	2^{-29}		radians
$\sqrt{A}$	32	2^{-19}		$\sqrt{\text{meters}}$
t_{oe}	16	2^{4}	604,784	seconds
C_{ic}	16*	2^{-29}		radians
Ω_0	32*	2^{-31}		semi-circles
C_{is}	16*	2^{-29}		radians
i_0	32*	2^{-31}		semi-circles
C_{rc}	16*	2^{-5}		meters
ω	32*	2^{-31}		semi-circles
$\dot{\Omega}$	24*	2^{-43}		semi-circles/sec
IDOT	14*	2^{-43}		semi-circles/sec

Figure 5: Ephemeris parameters from the IS-GPS-200

Where:

$$l_i^{(k)}(\vec{r}_{i0}) = \sqrt{(X_{sv} - x_{i0})^2 + (Y_{sv} - y_{i0})^2 + (Z_{sv} - z_{i0})^2} \tag{70}$$

$$\vec{e}_i^{(k)} = \frac{1}{\sqrt{(X_{sv} - x_{i0})^2 + (Y_{sv} - y_{i0})^2 + (Z_{sv} - z_{i0})^2}} \begin{bmatrix} (X_{sv} - x_{i0}) \\ (Y_{sv} - y_{i0}) \\ (Z_{sv} - z_{i0}) \end{bmatrix} \tag{71}$$

Rewriting equation (69) in control and estimation nomenclature we obtain:

$$z^{(k)} = z_0^{(k)} - (\vec{e}_i^{(k)})^T r_i \tag{72}$$

Where:

$$z^{(k)} = l_i^{(k)}(\vec{r}_{i0}) \tag{73}$$

$$z_0^{(k)} = l_i^{(k)}(\vec{r}_{i0}) + (\vec{e}_i^{(k)})^T \vec{r}_{i0} \tag{74}$$

Substituting back the linearized version into (17),

$$\rho_{IF}^{(k)} - \rho_{IF}^{(r)} - (z_0^{(k)} - z_0^{(r)}) = ((\vec{e}_i^{(k)})^T - (\vec{e}_i^{(r)})^T) \cdot r_i + m_\rho^{(k)} - m_\rho^{(r)} + \nu_\rho^{(k)} - \nu_\rho^{(r)} \tag{75}$$

$$\lambda\Phi_{IF}^{(k)} - \lambda\Phi_{IF}^{(r)} - (z_0^{(k)} - z_0^{(r)}) = ((\vec{e}_i^{(k)})^T - (\vec{e}_i^{(r)})^T) \cdot r_i + m_{\lambda\Phi}^{(k)} - m_{\lambda\Phi}^{(r)} + \nu_{\lambda\Phi}^{(k)} - \nu_{\lambda\Phi}^{(r)} + \lambda N^{(k)} - \lambda N^{(r)} \tag{76}$$

This expression now can be written in terms of the Matrix H:

$$\begin{bmatrix} \rho_{IF}^{(k)} - \rho_{IF}^{(r)} - (z_0^{(k)} - z_0^{(r)}) \\ \lambda\Phi_{IF}^{(k)} - \lambda\Phi_{IF}^{(r)} - (z_0^{(k)} - z_0^{(r)}) \end{bmatrix} = \begin{bmatrix} (\vec{e}_i^{(k)} - \vec{e}_i^{(r)})^T & 0 & 0 & I & 0 & 0 \\ (\vec{e}_i^{(k)} - \vec{e}_i^{(r)})^T & 0 & 0 & 0 & I & I \end{bmatrix} x_a \tag{77}$$

Recalling that $x_a = [r_i, b_a, b_w, m_\rho, m_{\lambda\phi}, \lambda N]^T$. As can be seen, a reduced form of the stochastic processes is chosen. This approach assumes the difference between each multipath state and the reference is a single state, to reduce the amount of total states of the system. The same argument is done for the Integer Ambiguity states.

Note that the unit vectors $\vec{e}_i^{(k)}$ need to be expressed in the same reference as r_i for consistency with all the calculations.

4.1.2 Covariance of the System

The first step to being able to apply a Kalman filter is to know the statistics of all the inputs and states of the system. This is done through the computation of the covariance matrix which contains all the necessary information to describe the process uncertainty.

The covariance of a vector (more precisely, autocovariance) can be written in the following form:

$$Cov(w) = \mathbb{E}(w \cdot w^t) \tag{78}$$

Where $E(\cdot)$ is the Expected value operator, i.e. the mean. To apply this expression, w needs to be a process with 0 mean, meaning that to compute the covariance of the system dynamics we will substract the mean value. Noting the difference to the mean as $(w - \bar{w}) = \delta w$, the measurement model, reduces to:

$$\delta z = H\delta x_a + \nu \tag{79}$$

The covariance of z is (Note that x_a and ν are uncorrelated):

$$Cov(z) = \mathbb{E}((H\delta x_a + \nu) \cdot (H\delta x_a + \nu)^t) = \mathbb{E}(H\delta x_a \cdot \delta x_a^t H^t) + \mathbb{E}(\nu \cdot \nu^t) \tag{80}$$

$$Cov(z) = HPH^t + R \tag{81}$$

In which P is defined as the covariance of the state error δx_a and R the covariance of the measurement.

If we follow a similar procedure for the mechanization equations, we will be able to obtain:

$$P_{k+1} = \Phi P_k \Phi + Q_d \tag{82}$$

For details on the demonstration of this particular expression please refer to [3]. In this expression all the Q_d terms have been grouped into a single expression because it can be computed with a simple numerical algorithm in matrix form.

The Van Loan Method allows for the computation of the discrete Power Spectral Density of the inputs Q_d given it's continuous counterpart Q. This method was developed to compute the discrete equivalent model from a given continuous LTI and is of great use due to its formulation simplicity and number of steps

This method consists of 2 steps. Firstly, Creating the matrix A:

$$A = \begin{bmatrix} -F & GQG \\ 0 & F^t \end{bmatrix} \Delta T \tag{83}$$

Secondly, computing the exponential of the A matrix will result in the following matrix:

$$e^A = \begin{bmatrix} s_1 & \Phi^{-1} Q_d \\ s_2 & \Phi^t \end{bmatrix} \tag{84}$$

From which Φ and Q_d are easily obtained. s_1 and s_2 are of no concern for our application.

As a consequence, in order to characterize the uncertainty of the system we need to compute the initial covariance of the state x_a, P_0; the power spectral density of the input noise Q and the covariance of the white noise of the measurement vector R.

With this quantities determined, we will be able to propagate all other noise estimates needed for the computation of the Kalman Gains.

Section 4.3 will focus on the measurement of this quantities for all the different processes involved in the formulation.

4.2 Monitor

Lastly, the detector itself will be explained. Detecting Spoofed GPS signals, which are indistinguishable from signals broadcasted by satellites, is a complex problem that focus of huge amount of research. This detection is usually accomplished through the use of extra auxiliary sensors to provide redundancy.

This monitor however, uses the standard INS and GPS sensors, widely used in all vehicle applications in the whole industry to detect this perturbations. Comparing the IMU estimates to the GPS estimates it is possible to assess the veracity of the signals.

To measure this discrepancy, the monitor implemented uses the Kalman innovations as the metric. An innovation based monitor with a tightly coupled GPS scheme will provide enough information to detect certain kinds of faults.

A smart spoofer might use his knowledge of estimation systems to introduce a fault that is small at the beginning, with the aim of corrupting the INS states, increasing up the velocity as the estimation of the vehicle loses quality. This type of fault is much more dangerous that any instantaneous fault that doesn't corrupt the states over time. In this report a detector for the harmful cumulative faults will be developed.

To account for this type of faults, the monitor used in this implementation will accumulate the normalized innovations γ_i, which are the residuals of the Kalman filter estimates.

The test statistic used will have the form:

$$q_k = \sum_{i=0}^{k} \gamma_i^T S_i^{-1} \gamma_i \tag{85}$$

In this formulation, S is the covariance matrix of the innovations computed in the Kalman filter algorithm. q_k is therefore a normalized innovations test statistic with a known distribution.

$$S_k = H_k P_k H_k^T + R \tag{86}$$

It is important to note that the Kalman innovations are independent ($\mathbb{E}(\gamma_i \gamma_j^T) = 0$ from a measurement epoch to the other [4] for both loosely and tightly coupled schemes.

The statistic is then compared to a predefined threshold T and a detection alarm is set any time the test statistic is greater that the threshold $q_k > T$.

$$q_k \lessgtr T \tag{87}$$

4.2.1 Monitor statistics

Under fault free conditions, the innovation vector γ_i follows a Gaussian distribution $N(0, S_i)$. The test statistic, q_k will therefore follow a chi-square distribution of nk degrees of freedom for the tightly coupled implementation (n being the length of the innovation vector). For a given false alarm, the test statistic follows a non-centrally chi-square distribution with a noncentrality parameter of:

$$\lambda_k^2 = \sum_{i=0}^{k} \mathbb{E}[\gamma_i^T] S_i^{-1} \mathbb{E}[\gamma_i] \tag{88}$$

To find the detection threshold T the inverse chi-square distribution is used and a probability of false detection needs to be defined. The smaller the false detection probability the less sensitive the filter will be. Similar to all the statistical detectors, a trade of situation needs to be resolved. In this report, we will choose a risk criteria of 10^{-2} for all the experiments

4.3 Data Analysis

Once all the models have been described and the general procedure of estimation is outlined, a quantitative analysis of the sensors and data is needed. This analysis will consist of the statistical characterization of all the stochastic processes and models and will include comparisons to the manufacturers data-sheets and expected performance.

The data analyzed in this section will be used in the model's covariance matrix for the use in the Kalman Filter estimation. Firstly we will discuss the computation of the Q and R matrix, and later the initial covariance P

Since Q contains the white noise from the IMU as well as the driving noises for the time varying biases, the IMU statistics will be discussed first.

Its important to note that the precision of this estimates will determine the performance of the detector. Overestimating the noise estimations will result in degraded performance since we would be purposely introducing modelling errors in the algorithm. To avoid such degradation, careful modeling is needed and measuring the actual data to be analyzed is highly recommended.

4.3.1 IMU statistics

The market for IMU sensors distinguishes different grades of sensors, each grade with a different performance and price. This performance is measured in how small the noises in the signals are. The different grades that can be purchased range from the lowest performance at Automotive grade, to Industrial, Tactical and Navigation, with the highest performance. The cost of each IMU also depends on its grade,a Navigation grade IMU can cost well over $100000. This cost can go down to a few dollars per chip for simple low quality IMU sensors that are not needed for high precision tasks.

In this report's implementation, The IMU of choice is the STIM300 from Sensonor. This model, characterized as a low Tactical grade IMU, is a MEMS

sensor containing and accelerometer, gyro and inclinometer in the same low profile package.

For the implementation only the accelerometer and gyro will be used and characterized.

The values usually reported in the IMU datasheet usually measure the characteristics of the integrated signal. It is common for a manufacturer to report the Random Walk parameter of the gyro or the accelerometer (Angular Random Walk or Velocity Random walk respectively).

The Angular Random Walk (ARW) of the STIM300 given by the manufacturer is $0.15\,\mathrm{deg}/\sqrt{hr}$. This value is an indication of how fast the angle measured from the IMU's gyro integration will drift over time. In particular it is a measure directly related to the standard deviation of the signal.

If the gyro signal is integrated to measure position, the expected variance of the angle estimation is expected to grow following a Random Walk model. Written in discrete form, the standard deviation of the angle will be:

$$\sigma_\theta = \sigma_w \sqrt{\Delta t} \sqrt{t} \tag{89}$$

Here, σ_w is the standard deviation of the gyro signal and Δt is the sampling time. Grouping together these 2 terms into 1 and we obtain the Random Walk parameter.

$$ARW = \sigma_w \sqrt{\Delta t} \tag{90}$$

if the standard deviation of the signal is known, the RW parameter can be easily computed, which in turn can be used to characterize the noise performance of the estimator.

The same procedure is done for the accelerometer standard deviation, this time measuring the velocity instead of the angular orientation. The Random Walk parameter of the accelerometer is called Velocity of Random Walk by analogy to the gyro.

For this particular IMU, the expected values for the ARW and VRW measured from their standard deviation are the following.

Sensor	Standard deviation	Random Walk
Accelerometer	$0.019044 m/s^2$	$1.703 \cdot 10^{-3} m/s/\sqrt{s}$
Gyro	$0.001323\,\mathrm{deg}/s$	$1.183 \cdot 10^{-4}\,\mathrm{deg}/\sqrt{s}$

Table 1: Random walk estimation from manufacturer's data

These values can be measured by sampling the sensor for short periods of time each run, to avoid variations in the slow time depending First Order Gauss Markov process of the bias. Since the correlation time of the process is big, close to an hour, measuring shorts periods of time allows to consider the process locally constant and subtracted by averaging the signal.

This parameters then can be used to compute the Power Spectral Density of the noise, which in case of the Random White Noise is simply:

Sensor	Bias Instability	PSD
Accelerometer	$4.9033 \cdot 10^{-4}$	$4.8085 \cdot 10^{-12}$
Gyroscope	$2.4241 \cdot 10^{-6}$	$1.1752 \cdot 10^{-16}$

Table 2: PSD values for the driving noises for the sensors' biases

$$PSD_{gyro} = ARW^2 \tag{91}$$

$$PSD_{accel} = VRW^2 \tag{92}$$

From this expression we obtain the first 6 terms of the PSD matrix Q, 3 for the 3 axis of the accelerometer and 3 for the gyroscope. We assume that the noise is uncorrelated between axis.

The next step is to model compute the PSD of the driving noises governing the bias drift of the sensors.

The bias are modeled as First order Gauss-Markov processes, which is known to have a closed form for the dynamics of its covariance. This dynamics are described with a time varying Lyapunov equation as:

$$\dot{P} = \left(\frac{-1}{\tau_b}\right) P + P \left(\frac{-1}{\tau_b}\right)^t + Q \tag{93}$$

For a one dimensional Gauss-Markov process in steady state,

$$0 = \left(\frac{-2}{\tau_b}\right) \sigma_b^2 + q \tag{94}$$

$$q = \left(\frac{2}{\tau_b}\right) \sigma_b^2 \tag{95}$$

This expression allows to compute the driving noise if the covariance of the steady state process is measured. To estimate this value, the specification sheet of the STIM300 is needed.

The steady state standard deviation of the bias covariance is given by the datasheet entry "Bias Instability" which corresponds to the minimum value of the Allan variance curve for that sensor.

For this calculation, the unknown correlation time of the Gauss Markov process is needed. This value is usually measured through the Allan Variance plot as well. The time at which the minimum of the Allan variance curve occurs is usually a good estimate of the correlation time of the Gauss-Markov process. In the case of the STIM300 is approximately $10^5 s$ for both gyro and accelerometer.

This defines the known components of Q corresponding to the IMU noise processes which are then to be converted to their discrete version through the Van Loan Method.

The rest of components of Q are related to the processes of the GPS measurements and will be discussed in the following section.

4.3.2 GPS Statistics

As discussed in the previous GPS sections, the noise in GPS signals will be modeled as three different stochastic processes. The First Order Gauss-Markov, random constants and random white noise. First we will focus on First Order and Random Constants to completely characterize the Q matrix and then the White noise will be studied to measure the R matrix.

In this GPS implementation, modeling the noises is not as straightforward as in the case of the IMU.

First we will study the multipath. Each satellite has a perturbation related the the amount of reflection of the signal received from the satellite. This amount of reflection changes with the environment, It is not expected to have the same multipath characteristics driving along an urban canyon as in flying through the air in a plane, where there are basically no obstacles for the signal to reflect on.

Therefore, the model of the multipath will be different depending on the environment. In order to keep the generality of the modelling, in the estimation algorithm, a multipath model will be introduced in the modeling but the first experiments and tests in this report will be conducted with signals from a GPS simulator, without multipath and atmospheric delays.

It is left for future experimentation the effects of multipath, tropospheric and ionospheric models in the detection performance.

For this reason, the model is set up analogously to the IMU's time varying biases, but the time constants and standard deviations would be selected accordingly to the geography and environment.

In this report, a nonzero time constant of $100s$ is selected with a Standard deviation close to 0, to ignore the effects of multipath.

There is a subtlety in the computation of the covariance due to the specific model that we selected. This effect is worth mentioning since it also applies to the Integer Ambiguity states.

The measurement model selects a satellite as a reference from which to measure the differences, which will correlate all the noises in the process. This implies that the PSD sub-matrix for multipath and Integer ambiguity is not diagonal as in the IMU.

To compute the actual PSD matrix, we begin computing a PSD matrix with the values of each different satellite in the diagonal and 0 elsewhere. This matrix Q_i' will be different for the code and carrier signals but always diagonal.

Now to account for the correlation between signals of the actual model, we define the Single Difference matrix. This matrix takes a raw set of measurements of code or carrier and outputs the Single Difference model with one of the satellites as the reference.

As an example, if the number of satellites was 5 and the 2^{nd} is selected as the reference satellite, the transformation would have the form:

$$T_{SD} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{96}$$

This matrix computes the following transformation.

$$z_{SD} = T_{SD} \cdot z_{raw} \tag{97}$$

Therefore it can be showed that for the PSD matrices, if the Raw Measurement covariance is known, the Single Difference can be computed simply by the following equation.

$$Q_{SD} = T_{SD} \cdot Q_i' \cdot T_{SD}^t \tag{98}$$

Using the precomputed matrices of the multipath driving noise covariance, the PSD of the Single Difference can be computed easily.

This approach can also be applied to the Integer Ambiguity states. This uncertainty is modeled as a Random constant, which generally has no noise input since the model is simply:

$$\lambda \dot{N} = 0 \tag{99}$$

This approach however is not recommended when implementing a Kalman filter. The mathematical properties of random constants are not favorable for the set of matrix operations that need to be done in the Kalman Filter. The consequences of ignoring process noise might cause the filter to be unstable, which is unacceptable.

Another reason to add process noise in the states is the following. With no noise input, the covariance of the ambiguities will decrease at each measurement epoch without any bound. This might seem positive but when working with real data, with the models only approximating the complex reality, it might mean that the state converges to a biased estimate due to the uncertainty in the initial conditions and lead to a biased filter overall.

To account for the effect, a process noise is added to the state that will make the uncertainty grow in the propagation, allowing the filter to adjust the value according to the new data.

The procedure to define the sub-matrix Q_N is the same as for the multipath, here selecting a driving noise of $0.005m$ and assigning it to each of the satellites, to then transform it to the Single Difference Model.

Summarizing he PSD matrix Q will have the form:

Noise	Standard deviation
Code	0.05 m
Carrier	0.002 m

Table 3: Thermal noise of the simulated GPS data

$$Q = \begin{bmatrix} PSD_{arw} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & PSD_{vrw} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & PSD_{ba} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & PSD_{bw} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & PSD_{SDmp}^{(code)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & PSD_{SDmp}^{(carr)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & PSD_{SD}^{(\lambda N)} \end{bmatrix} \tag{100}$$

Where the matrices PSD_{arw}, PSD_{vrw},PSD_{ba},PSD_{bw} are 3×3 diagonal matrices and $PSD_{SDmp}^{(code)}$,$PSD_{SDmp}^{(carr)}$,$PSD_{SD}^{(\lambda N)}$ depend on the total amount of satellites in the observation, if n_{sat} is the number of satellites, this matrices will have size $(n_{sat} - 1) \times (n_{sat} - 1)$.

The next step to characterize the statistics of the data is to estimate the measurement matrix R, the white noise process noise of the measurement model. In this case, being a Single Difference model, the white noise is internally correlated as described before, which makes the characterization similar to the previous signals.

GPS white noise is usually referred as thermal noise, since a great part of the noise is due to the thermal photons radiated from all bodies due to their temperature. In the case of our simulated data, measuring the thermal noise is straightforward, since the data does not have multipath and atmospheric delays.

In this report the values for thermal noise in the GPS data are shown in Table 3

Since data in the measurement equation does not need to be converted to PSD, there is no further processing needed in the data and the R matrix can be written as:

$$R = \begin{bmatrix} T_{SD}\Sigma^2_{code_{th}}T^t_{SD} & 0 \\ 0 & T_{SD}\Sigma^2_{carr_{th}}T^t_{SD} \end{bmatrix} \tag{101}$$

Here, $\Sigma^2_{i_{th}}$ is the diagonal variance matrix of the given signal i thermal noise.

4.4 Initialization

Once the system is defined and it is ready to test, the first step will be initializing the IMU with plausible states and find the initial covariance of the state vector.

The process of Initialization is critical for the performance, especially if the mechanization is computed without GPS aiding. In submarines, for example,

GPS signals do not reach far below water, which makes impossible the use of GPS. It is only when they resurface that the calibration is possible and the actual position of the ship can be found. During the underwater journey, they integrate the IMU signals with help of some other aiding sensors like magnetometers. As a consequence, submarines need the highest grade IMU's.

In order to acquire good estimates for the initial values of the state and covariance matrix, the next procedure is followed at start-up.

- **Bias Estimation**: The first step for bias estimation is the computation of the IMU initial biases. This is achieved by averaging several seconds of data received from the IMU.

  This provides an estimate for the biases that prevent the signals from diverging initially. After a few seconds, the biases will drift and feedback will be needed to keep the divergence stable.

- **Initial Attitude**: Next, the average accelerometer output can be used to estimate an initial attitude assuming that the IMU is static during the averaging period.

$$\phi_0 = \arctan_2(u_E, u_D); \tag{102}$$

$$\theta_0 = \arctan_2(-u_N, \sqrt{u_E^2 + u_D^2}) \tag{103}$$

  Following the expressions, the initial pitch and roll of the IMU can be found. All except yaw, which is unobservable unless an aiding sensor like a magnetometer is used. (Note that the 4 quadrant arctan function is used to compute the inclination and the sign without uncertainty).

- **Initial position**: Find an approximate solution of the initial position, a simple Least Squares estimator for position is executed until convergence. The position will be used as the initial estimate of for the position in the Kalman filter estimator. This estimation is precise to a several meters and thus it is useful as an initial value even though the precision is low.

- **Initial Covariance**: Lastly, it is possible to estimate the initial covariance of the states. The states whose covariance needs to be set are: position, velocity, biases, multipath and integer ambiguity. The position's covariance can be set to $(5m)^2$ value as a conservative estimation of the precision of the Least Squares estimator.

  Since the initialization period needs the IMU to be at rest in order to average, the error in velocity can be set low initially, in the order of $(0.05m/s)^2$.

  The covariance for the Euler angles can be computed exactly thanks to the algorithm we used for their estimation. The algorithm in question is called "Gyrocompassing" and it is well known that the covariance of the attitude error will be related to the accelerometer bias. This is caused because the estimation of the measured gravity is the same as the estimation of the initial bias of the accelerometer.

The mixed covariance of the attitude vector and the bias states will be:

$$P_{Eb_a} = \begin{bmatrix} H_g P_{b_a} H_g^t & -H_g P_{b_a} \\ -P_{b_a} H_g^t & P_b a \end{bmatrix} \tag{104}$$

Where:

$$P_{b_a} = \sigma_{InRun}^2 + \frac{PSD_{vrw}}{T_{init}} \tag{105}$$

$$H_g = \begin{bmatrix} 0 & \frac{1}{u_D(u_E^2/u_D^2+1)} & \frac{-u_E}{u_D^2(u_E^2/u_D^2+1)} \\ \frac{-\sqrt{(u_E^2+u_D^2)}}{(u_N^2/(u_E^2+u_D^2)+1)} & \frac{u_N u_E (u_E^2+u_D^2)^{3/2}}{(u_N^2/(u_E^2+u_D^2)+1} & \frac{u_N * u_D (u_E^2+u_D^2)^{3/2}}{(u_N^2/(u_E^2+u_D^2)+1} \\ 0 & 0 & 0 \end{bmatrix} \tag{106}$$

Note that the In Run Variance is a measure of the variance between different start-ups given by the manufacturer in the specification sheet.

For the code and carrier multipath signals, we choose the steady state variance used to compute the driving noise.

Lastly, for the Integer Ambiguity we will choose an initial variance of $(2m)^2$ for the reason discussed previously in Section 4.3

Initialization ends with the generation of the system matrices from the values computed previously.

4.5 INS Only Performance

With all the necessary information prepared, we can begin the experimentation. To evaluate the modeling of the mechanization, IMU and noise parameters, the first experiment will consist of an static scenario with no actual acceleration.

In this experiment, we will not use the Kalman filter update to correct the IMU signals, which will allow the assessment of the design choices and the noise covariance models.

Since the nominal trajectory of the IMU is known (Static), a Linearized Kalman filter has been chosen for the propagation. In a more general environment where the nominal trajectory is not know, a more general approach would be to update the matrices based on the data from the estimation, thus using an Extended Kalman filter.

The experiment consist en 10 runs of $20min$ each, to observe the statistical performance of the estimator.

Since the estimation is being performed without GPS corrections, the performance is not expected to be great. This results will be used as a comparison to the INS/GPS integration and to evaluate the statistics of the given noise inputs compared to the theoretical propagation of the covariance.

Beginning with the position estimates (Figure 6) In this results, the red line represents the 1-Sigma envelope given by the covariance estimates. Each run is represented with a different color. As can be seen the precision at $20min$ is close to $500m$, which is obviously intolerable for any kind of navigation purpose.

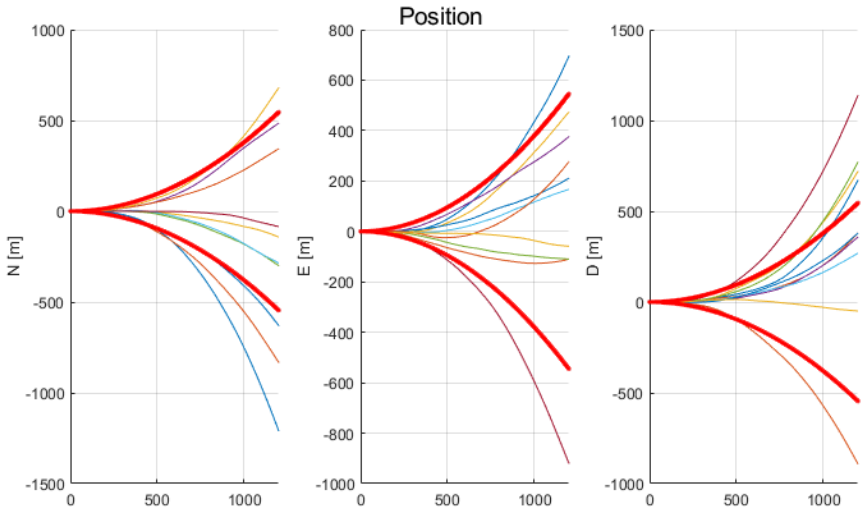


Figure 6: Position estimate of the Open Loop Propagation

Experimental data seems to follow the covariance envelope as expected (68% of the runs not crossing) if not better than what the covariance predicts.

The same argument can be done for the velocity (Figure 7), the runs seem to follow the expected spread closely. Note that the covariance envelope is almost linear instead of the expected square root rule for one dimensional IMU. This is due to the effects of the changing attitude, which introduce noise in the velocity estimate, degrading the performance. This downgrade is necessary to account for the small changes in orientation that a vehicle moving in 3D space might experience.

The same effect applies for the attitude estimation of Yaw, Pitch and Roll (Figure 8). What in 1D would be a simple Random Walk process, is corrupted in 3D by the value of the attitude itself, i.e. the rate of change of the attitude depends on the value of the attitude plus the gyroscope measurement, Equation (45). This adds an extra component to the covariance propagation that degrades the performance noticeably.

This effects justify the need of aiding sensors such as GPS and other. As a minimum, a magnetometer would be needed to give observations on the attitude, which would increase the precision of the estimates remarkably.

In this implementation we decided on the use of GPS as an aiding sensor to obtain feedback orientation on position directly and being able to further correct the IMU states.

4.6 INS/GPS Integration Performance

Now we will evaluate the performance of the system in closed loop, feeding GPS information to the INS. The GPS data arrives from the receiver at a rate of

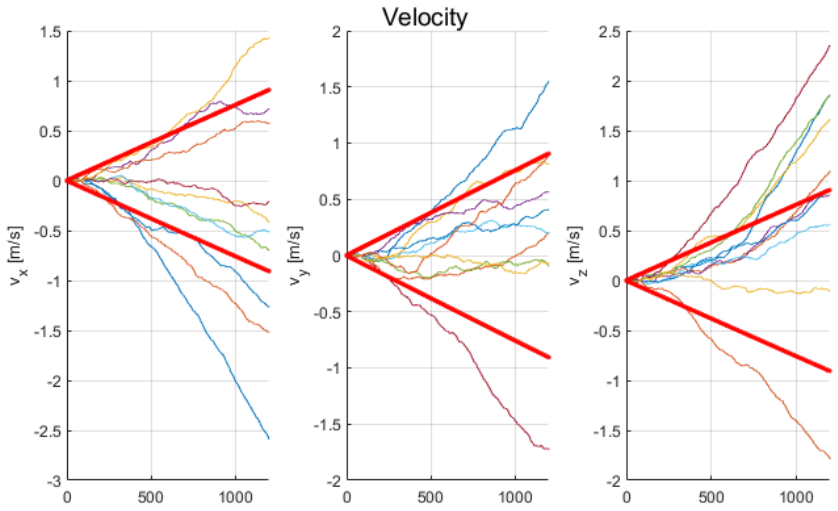


Figure 7: Velocity estimate of the Open Loop Propagation

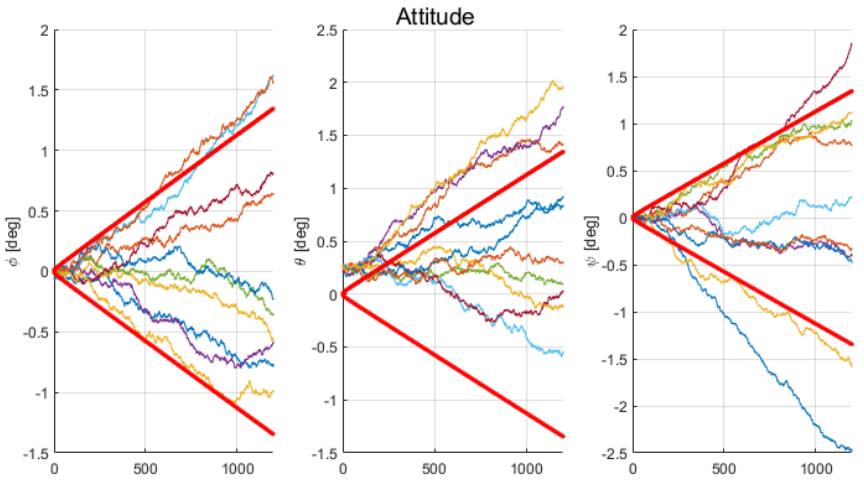


Figure 8: Attitude estimate of the Open Loop Propagation

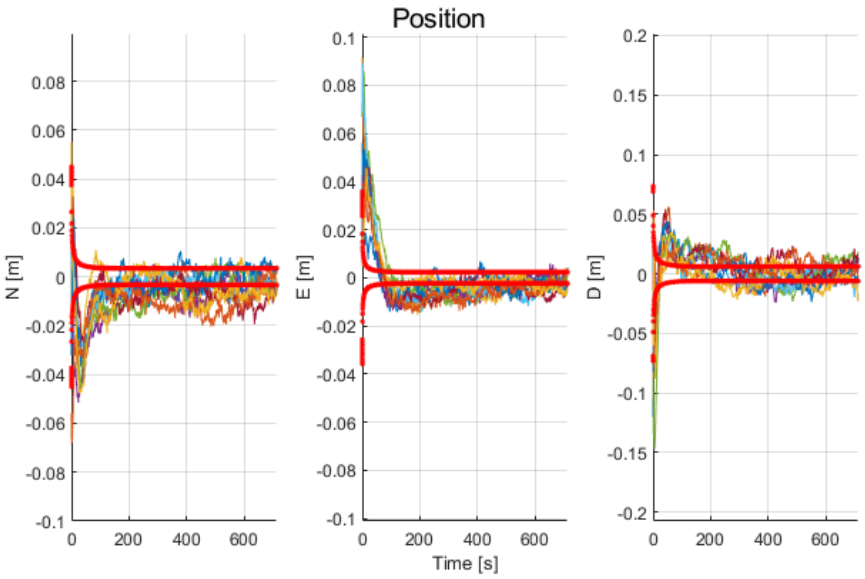


Figure 9: Position Estimates INS/GPS of Un-spoofed data

$2Hz$ and corrects the position using the Single Difference model described in previous sections.

This type of model belongs to the category of tightly-coupled GPS-INS systems. This kind of integration includes GPS states in the mechanization equations as opposed to just observing the position. Using this approach is more precise in terms of estimation but is more vulnerable to spoofing attacks.

We chose this measurement model not because the detection performance but because is a common type of integration in the industry, and the results would not be conclusive we used a simpler model like loosely-coupled GPS INS.

The following results have been calculated from the same data sets as in the previous section, to allow for a meaningful comparison between algorithms. As can be seen, the performance is substantially better, allowing for precise positioning (Figure 9). For our simplified case, the precision of the algorithm is $4mm$ at the end of the 10 minutes.

As expected, the GPS aiding outperforms the INS only estimation by orders of magnitude with stable estimates that converge to the real position of the system.

This however shows the vulnerability to spoofing of the system as a whole. Any type of bias in the GPS data will have repercussions on the estimation significantly. Is in this cases when the integration of the IMU in the system will prove remarkably useful.

5 Spoofing Scenario

Spoofing is the act of sending malicious information from a source other than the original. In this case, GPS spoofing consist in the broadcast of the GPS signals normally emitted by the satellites to trick the receiver into believing data that is not genuine.

A GPS receiver is unable of discerning between satellite data or spoofed data, since all the information is encoded in a similar fashion in both genuine and spoofed data. This is the reason why redundancy is needed to detect spoofers. In this research, the redundancy comes from the widely available inertial sensors, usually installed in the majority of vehicles instead of adding novel sensors or very specific hardware. The inertial sensors are usually added for autopilot, monitoring or safety reasons and are widely used in navigation.

From the spoofer point of view, there are different types of faults that differ in the amount of information needed from the vehicle.

For example, a simple jump fault. In this scenario the spoofer sends a GPS signal from another location to the receiver. The receiver measures the new signal as genuine and shifts the position to the new one. This fault does not need any previous information from the user but is easily detectable.

We will focus on accumulated faults. This faults shift the position slowly but continuously from the initial position, corrupting the IMU states progressively and injecting a erroneous position estimate.

This is noticeably more complex to detect, since the shift is never discontinuous and is small from one measurement to the next.

To measure this type of fault, we need an accumulated monitor that is capable of accounting for all these small shifts and detect the fault before it becomes excessively dangerous.

This kind of fault requires the spoofer to know precisely what the exact position of the vehicle at a given time is, to input a believable shift of position without being detected. As expected, this kind of fault is harder for the spoofer to set up and requires some more complex hardware to measure the position of the vehicle (Laser rangefinders for example).

We will test our monitor to a fault profile with these characteristics to measure its performance.

A third case of fault that requires the spoofer to know more information is the Worst Case Fault. This fault is computed to minimize the probability of detection but requires the spoofer to know everything about the vehicle, even the internal state of the estimates. This includes the algorithm used itself, the covariance matrices and the position of the vehicle at any given time.

Even assuming that the spoofer has perfect knowledge of the vehicle internal information and is capable of generating this fault, theoretically [4], this detector could detect it, proving its efficacy to even the hardest spoofing. To prove it experimentally, it has been devised a method to inject these faults into the estimator to test the practical performance.

The scenario for the tests will be similar in all the experiments. The experiment begins with a period of initialization, enough to allow all the estimates to

reach a steady state solution, from which the fault can be injected.

The experiment will begin with $10min$ of static measurements for initialization, followed by the spoofed shift. The shift will always be $10m$ with changing spoofing time.

Accounting for the fact that the vehicle will move out of the range of the spoofer's measuring system, the individual is limited to a finite amount of spoofing time, forcing the fault to be finite and limiting the spoofing capabilities.

Additionally the monitor will be tested against for pre-monitoring performance. This scenario accounts for the fact that the monitor might start measuring innovations some time before the actual fault begins, accumulating true information in the statistic.

This questions will be answered in the following sections.

5.1 Anti-Spoofing Performance

5.1.1 Steep fault

The first fault to be analyzed is composed of a static period of $10min$ for initialization and a shift East of 10 meters in 30 seconds. The fault is steep to demonstrate clearly the effects it has in the monitor.

The spoofer trajectory profile follows a power curve of fifth degree, starting slowly and increasing the acceleration as time progresses.

$$X_{east} = 10\left(\frac{t}{30}\right)^5 \tag{107}$$

Comparing that to the results from the state estimation in Figure 10. We can see that as expected, the position estimates for the North and Down are kept fairly constant while the East state drifts, as predicted by the spoofer.

But did we detect the fault? A functioning monitor will check the moving threshold against a cumulative statistic, both represented in Figure 11.

As we can see, 5 seconds from the beginning of the measurement, all runs have crossed the threshold and have been detected as malicious. At this point, corrective measurement could have been taken to prevent further drift in the estimates.

This easily detectable fault is of use to understand the mechanisms by which the detector is able to discern between spoofed and genuine data.

The monitor measures the sum of the normalized innovations over a period of time. From [4] we know that the innovation vector of a spoofed signal depends on the fault component added to the measurement. If the fault f added is written as follows, the innovations will have the form of equation (109).

$$z_k = Hx_k + \nu + f_k \tag{108}$$

$$\mathbf{E}(\gamma_k) = \mathbf{E}(z) - \mathbf{E}(Hx_a) + f_k \tag{109}$$

As can be seen a fault injected in the measurement shifts the innovations away from their expected value. This result is obvious when looking at the

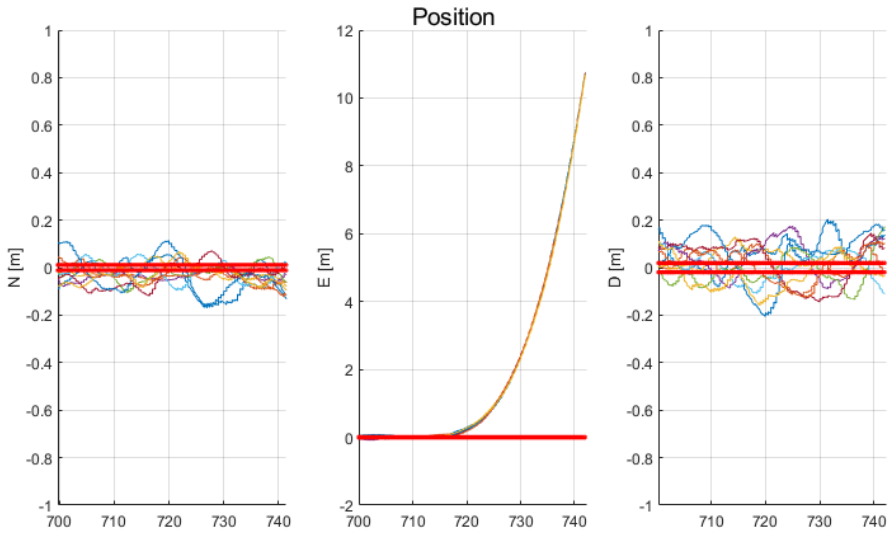


Figure 10: Position estimates for Spoofed 10m in 30s

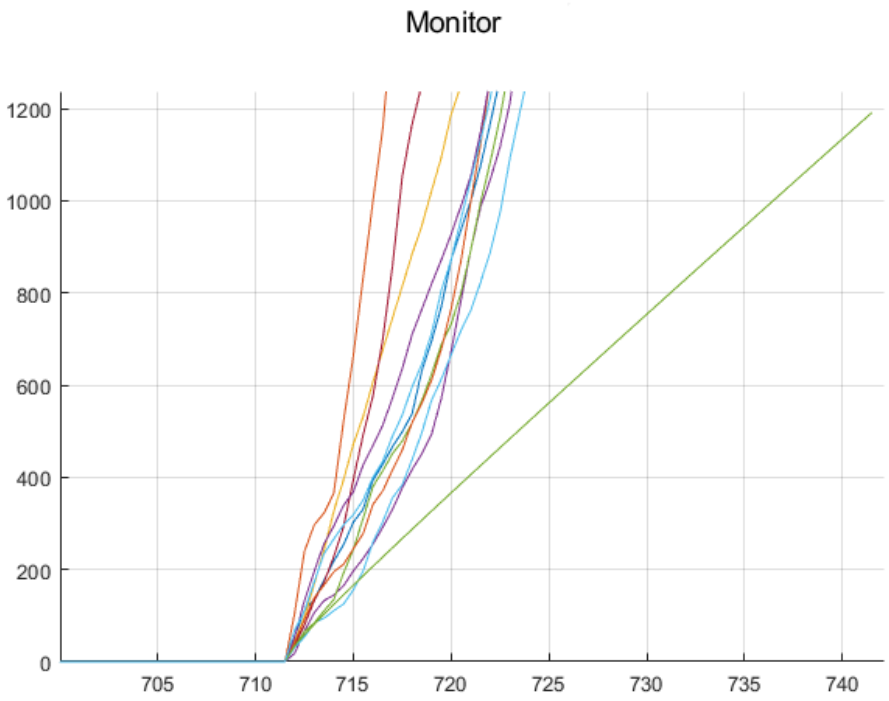
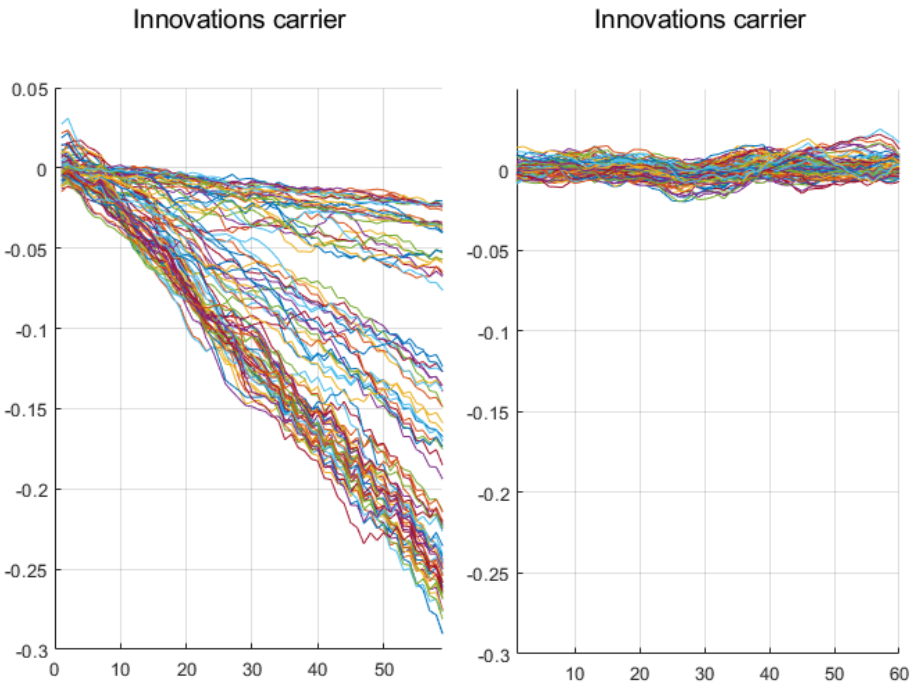


Figure 11: Monitor statistics for a 30m in 10s fault, in green the threshold



(a) Spoofed carrier innovations.

(b) Un-spoofed carrier innovations.

Figure 12: Comparison between carrier innovations for spoofed and un-spoofed experiments.

innovations directly in Figure 12. This figure shows the comparison between spoofed and un-spoofed innovations and the effects of the fault are clear. The shifts are then added to the statistic (which squares the innovations, equation (85)) until it surpasses the threshold.

The velocity at which detection is achieved will mainly depend on how steep the fault is and therefore, slow and small faults carry more risk than short and large ones.

5.2 Performance Evaluation

Now that all the internal behaviour is clear for both unspoofed and spoofed data, we will proceed to the evaluation of the performance to different fault scenarios.

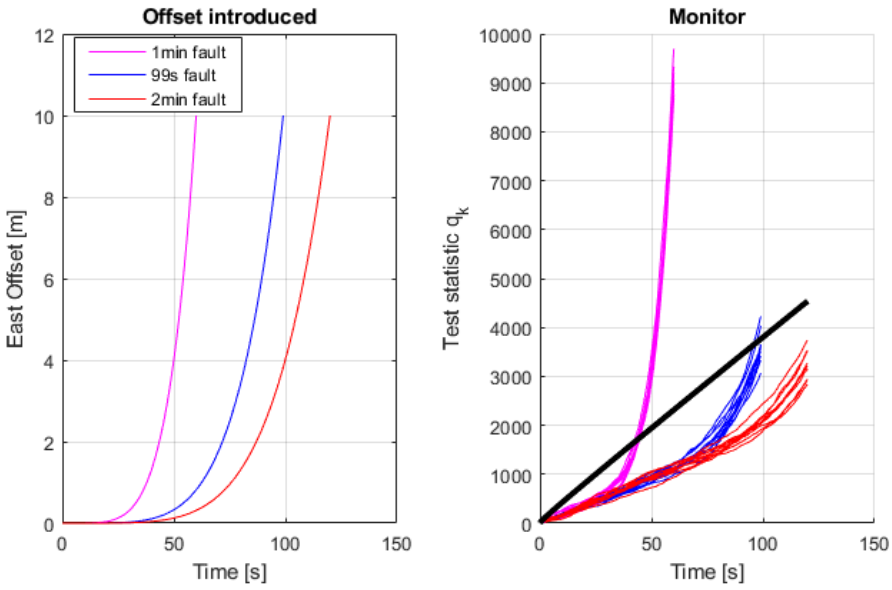


Figure 13: Sensitivity to spoofing time

5.2.1 Sensitivity to spoofing time

First of all we will evaluate how the allowable time to spoof the estimator affects the monitor.

Intuitively, the more time the spoofer has to inject a given fault the more dangerous it is for the estimator due to state contamination. State contamination is the process by which slowly injecting a fault changes the values of the state estimates of the vehicle without triggering detection.

The spoofer will want to have as much time as possible to inject the fault and do it the slower the better. This will ensure that the spoofing is not detected and the fault is introduced optimally. However the spoofing time will be limited due to the particular scenario or the duration of the precision maneuver.

But how much time would he need to actually inject the fault without being detected?

To find an answer, a spoofing scenario consisting of a 10m fault was devised. From a resting steady state position, an East drift was injected into the estimator GPS. The 10 meter fault has been introduced at different rates ranging from 1 to 2 minutes. Figure 13 showcases the results.

As can be seen any fault injected in less than a minute will have a detection rate of 100%. Is not until the spoofing time reaches $99s$ that the detection rate drops below total detection. After that point, as expected, a slow fault would be undetectable. This however shows that with this estimator and under a fault like the explained, a vehicle would be protected 1.5 minutes from spoofing with a 100% detection rate.

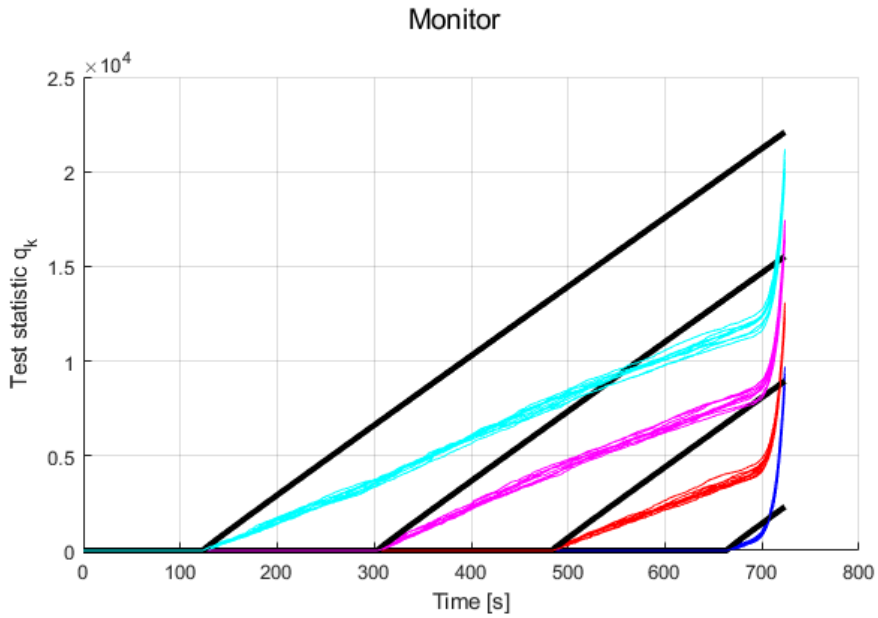


Figure 14: Pairs of Threshold and q_k to evaluate pre-monitoring performance

In this time, the vehicle could perform a critical maneuver like a landing or an approach comfortably and still be protected from attackers.

This experiment proves the spoofing time hypothesis by confirming that the spoofing risk increases with time. However, the time necessary to lower the detection rate would be enough to perform precision maneuvers thus ensuring the safety of the vehicle to the given specification.

5.2.2 Pre-monitoring performance

Pre-monitoring is the difference between the time at which the monitor is turned on and the time that the actual fault begins. Studying the effects that this difference may cause in the detector's performance sheds light into how useful the detector is when the time at which the signals will be spoofed is not known.

Since this is the most common case of spoofing, a monitor highly sensitive to this discrepancies is not reliable. Using a monitor with high sensitivity might lead to a quick degradation of the detection rate and the safety of the maneuver.

In the case of the tightly integrated innovations based monitor, this sensitivity is low compared to the sensitivity in attack time.

In Figure 14, the results of next experiment are represented. With a constant fault of $10m$ in $1min$, different pre-monitoring times have been tested. Starting from a concurrent fault with with a blue line, and retarding the monitoring time 3 minutes per set of data (Color).

For this particular fault, it can be seen that no performance is lost until in

the first 6 minutes, but it's at 9 minutes of pre-monitoring when the monitor looses detection capabilities.

With a Safe Time of more than 6 minutes, this monitor can be used to safely detect faults that would otherwise pose a serious threat to the integrity of the vehicle.

5.2.3 Spoofing sensitivity to measurement errors

All the previous section showed results for the ideal case that the spoofer has perfect knowledge of the position of the vehicle. This case however is usually unrealistic, error and noise in the measurement of the vehicle position is always expected.

This error might arise from the use of sensor to measure the position, like laser ranging sensors or optical sensors. This estimating error hinders the spoofer's capabilities which in turn increases the confidence in the estimator.

Theoretically [4], a noise in the order of a few millimeters is enough to increase the detection substantially. In this research, a noise of 3mm has been introduced to the GPS spoofed signals and compared to the same ideal case.

This simple addition of such a small noise in GPS terms, produces detectable faults that were previously undetectable. As an example, a 10m in 2 minutes fault, which was undetectable until now, when contaminated with 3mm white noise, the detection rates reaches 90% (Figure 15)

The Figure reflects the difference between the ideal and the contaminated signals. They are practically indistinguishable from each other. On the other side, this detector is able to distinguish between the two signals clearly and increase the detection rate in even the hardest faults.

Measuring the position of the vehicle to millimeter accuracy would be troublesome for the spoofer without using advanced measurement techniques, which protects the vehicle from a wide range of spoofing setups.

5.2.4 Worst Case fault

The last case that will be analyzed in this report is the behaviour of the monitor to the Worst Case Fault. This fault is the result of the following hypothetical scenario.

We assume there exists a spoofer with knowledge on the whole setup of the vehicle. The spoofer knows all the sensors, it's characteristics and even the estimation algorithm and the type of monitor that we use. With this setup he/she is capable of reproducing all the parameters of the navigation system, including position estimates and the monitor's behaviour. With all this information, it is possible to generate a fault that maximizes the following metric, integrity risk.

$$I_r = Pr(|\epsilon_k| > l, q_k < T) \tag{110}$$

The integrity risk is defined as the probability of a given fault exceeding a threshold value ($|\epsilon_k| > l$) while not triggering a monitor alert ($q_k < T$).

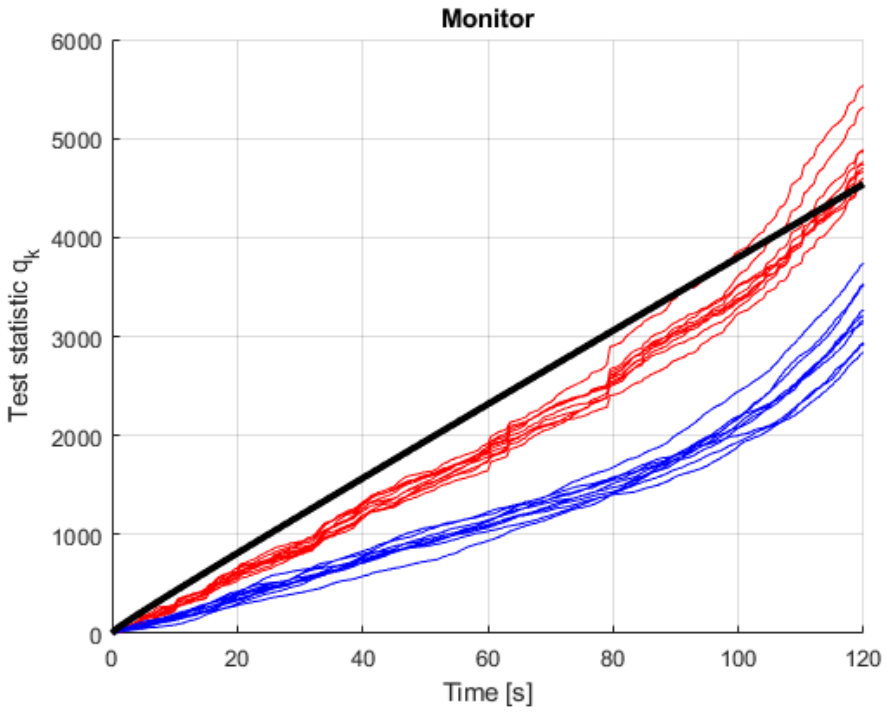


Figure 15: Effect of additive white noise in the spoofer estimation

Maximizing this metric using the data of our particular estimator is therefore generating the fault that has the maximum likelihood of not being detected while achieving a given deflection.

To generate the worst case fault we follow the derivation on [4], which provides the applicable fault vector f to be introduced in the GPS generator. The generator will then provide the generated signals to be introduced in the estimator.

Maximizing the Integrity Risk is equivalent to maximizing the failure mode slope ρ_k^2, defined as:

$$\underset{f_{1:k}}{\operatorname{argmax}}\, \rho_k^2 = \underset{f_{1:k}}{\operatorname{argmax}}\, \frac{\mathbb{E}(\epsilon_k)}{\lambda_k^2} \tag{111}$$

Writing this quantities in function of the fault profile $f_{1:k}$ allows for the computation of the worst case fault direction, written analytically as:

$$f_{1:k} \propto \bar{B}_{1:k}^{-1} S_{1:k} \bar{B}_{1:k}^{-T} A_{1:k}^T T_\epsilon^T \tag{112}$$

In this expressions, the matrices $A_{1:k}$ and $\bar{B}_{1:k}$ are aggregate functions of the system matrices. Each element of the A matrix can be written as:

$$A_{ik} = \begin{cases} (I - L_k H_k)\Phi(I - L_{k-1}H_{k-1})\Phi ... (I - L_i H_i)\Phi & \text{if i < k} \\ (I - L_k H_k)\Phi & \text{if i=k} \end{cases} \tag{113}$$

Similarly, the elements of matrix $\bar{B}_{1:k}$ can be written as:

$$\bar{B}_i = \begin{bmatrix} -H_k \Phi A_{1:k-1} & I & 0_{n \times n(k-i)} \end{bmatrix} \tag{114}$$

Computing the optimum fault magnitude reduces then to a one-dimensional search to maximize the Integrity Risk in function of the magnitude of the fault α through the generation of several samples of the measurement noise $\tilde{x}_i^s$.

$$I_r(\alpha) = \frac{1}{m} \sum_{i=1}^{m} Pr(|\epsilon_k| > l, q_k < T \mid \tilde{x}_i^s) \tag{115}$$

A spoofer capable of generating this type of fault requires a huge amount of information and background from the vehicle and only the most advanced of users could implement this faults. This is the reason why designing an algorithm that can detect even the worst of the faults available would prove itself to be a useful addition to the navigation scene.

The first fault to be tested will demonstrate the effect of the Worst Case Fault compared to unspoofed results. As can be seen in Figure 16, the difference between scenarios is minimal, the unspoofed results closely resemble the unspoofed data except at the beginning.

As expected, a spoofer with great knowledge of the system behaviour is capable of generating faults with very high integrity risk, posing a threat to the vehicle that cannot be allowed.

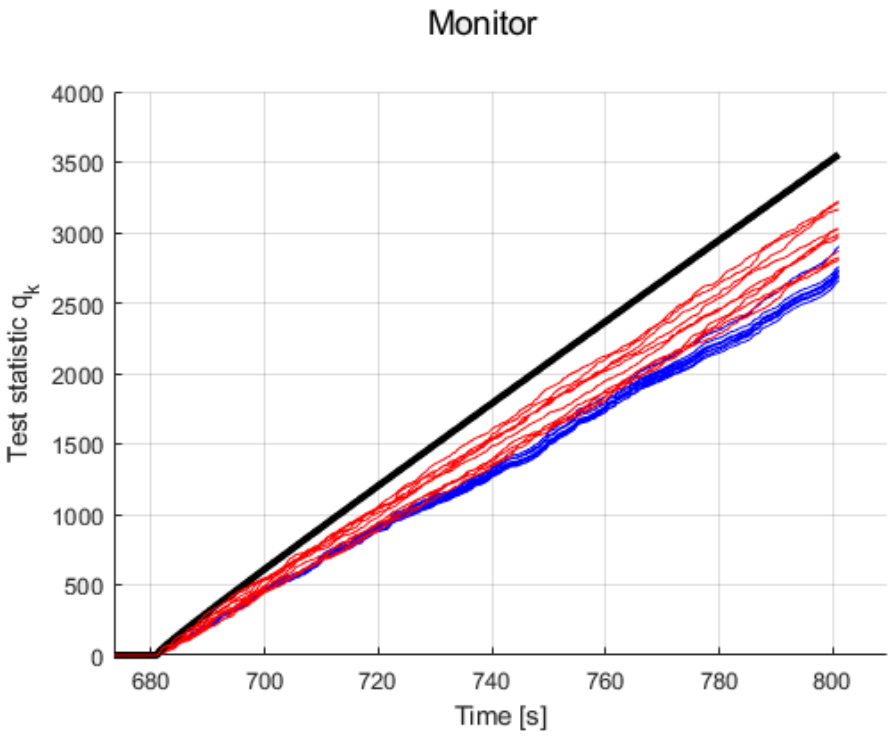


Figure 16: Worst Case Fault (red) against No fault (blue)

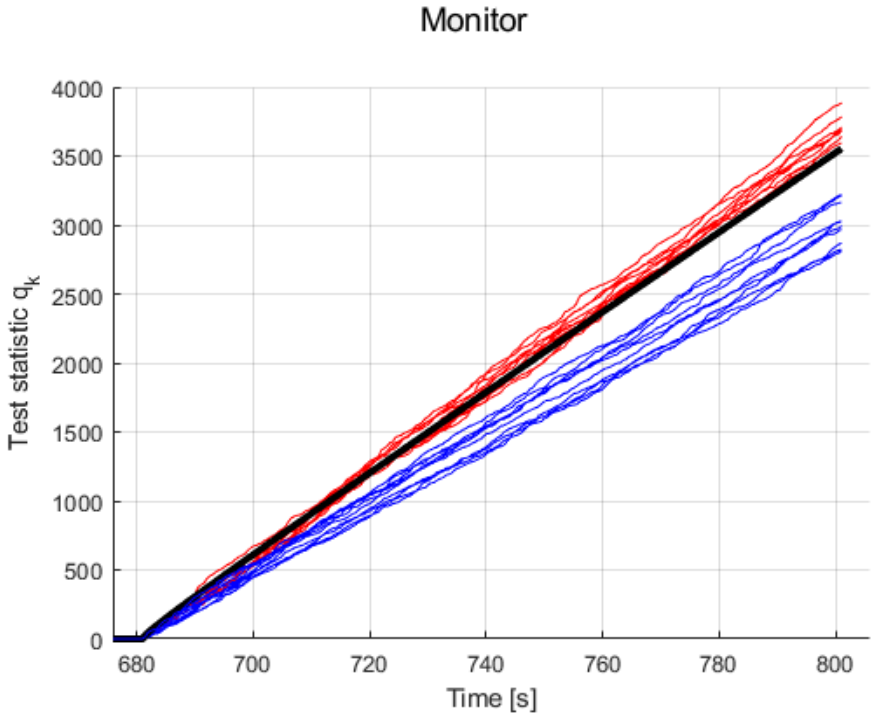


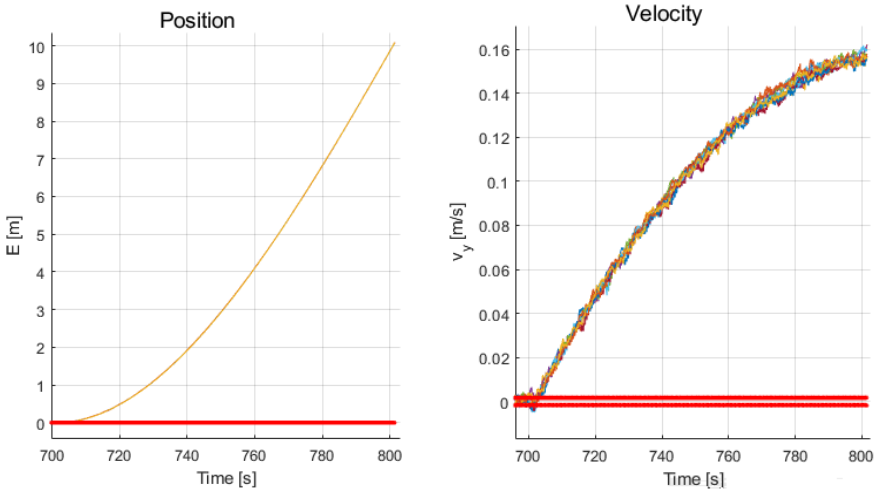
Figure 17: Worst Case fault with tracking noise vs ideal fault

However, we assumed that the spoofer is capable of tracking the position of the vehicle exactly, with absolutely no noise, which is unreasonable in a real time implementation. Since the spoofer would need some kind of positioning sensors to measure the position of the vehicle, those would add a white noise to the position measurements which would propagate to the worst case fault calculation. To be in the conservative side, we would assume a high end spoofing setup with quality sensors that only output a signal with a small white noise with $3mm$ of standard deviation.

Adding the white noise to the spoofer setup will generate a fault that is naturally noisy, creating a difference between the GPS expected noise and the actual signal. This increases the detection capabilities of the detection algorithm.

As can be seen in Figure 17, the simple addition of $3mm$ in the GPS estimations, allow the previously undetectable fault to be easily detected by the algorithm. The innovations based estimators are noticeably sensible to tracking noise and even a noise which is smaller than the actual standard deviation of the position states will trigger detection easily.

The Worst Case Fault, shown in 18, instead of increasing in velocity, it decreases it slowly over time. This can be attributed to the fact that a constant velocity fault is undetectable by a sensor that measures acceleration directly.



(a) Position estimates.

(b) Velocity estimates.

Figure 18: Estimations of the WCF spoofed estimator.

Only the changes in velocity can be detected, which the WCF optimizes to be the minimum possible.

As a result, the position estimates approach a straight line profile with minimum acceleration. This will minimize the faults in the innovations, promoting a more efficient fault.

Even though the Worst Case Fault is capable of reducing the detection rate of the algorithm significantly, any practical implementation of the spoofing algorithm, will prove highly troublesome if not impossible. The noise introduced due through the tracking will degrade the spoofing capabilities and will allow the detection of the fault.

6 Conclusions

The study conducted above analyses the performance of the Kalman filter estimation algorithm and the detection capabilities of the innovations sequence based INS monitor against GPS spoofing.

The algorithm was implemented in MATLAB successfully through the creation and testing of sub-programs easily testable to ensure a coherent functionality.

This algorithm was tested independently as an INS only propagation to examine the behaviour and the coherence between measured noise data and the numerical propagation. It was observed a high sensitivity to the initial noise parameters which led to a careful measurement of the data parameters instead of relying on manufacturer's data in the white noise estimates.

Then, the study of INS/GPS integration was evaluated. A tightly coupled scheme was selected to achieve high precision despite its disadvantages in spoofing. The precision and performance of the INS/GPS estimation algorithm improved upon only INS propagation. This grade of IMU is not high enough for an only INS integration and needs to be aided with GPS.

The following sections analyzed successfully the effects of spoofing in this system's performance to a variety of cases an faults. First the effects of an easily detectable fault were described to gain insight on the detection mechanism.

The performance under more complex faults and performance limits were then calculated. First, the effects of different spoofing periods in detection performance allowed to measure some time limitations of the algorithm. This limitations however were proved to be of low impact.

Following the spoofing time analyses, the study of pre-monitoring was conducted. The same fault was introduced on the system and it was proved that the protection obtained from the algorithm was significant for the applications, despite the assumed lack of knowledge in the spoofing starting time.

A more realistic scenario was then conducted, comparing the effects of the natural noise that a spoofer measuring system would introduce. It was proved that previously undetected signals were detectable only by the addition of a $3mm$ white noise to the spoofer's measuring system.

This study concludes with the computation of the worst case fault and the testing of the algorithm performance to the worst case of spoofing. An implementation of the worst case fault generation is outlined and implemented successfully to show its effects in the monitor.

This fault is then compared to the original unspoofed scenario to confirm the small differences between runs which confirms that the fault profile is significantly more harmful than the previous fault.

The last experiment of the report consists in the introduction of white noise to the spoofer tracking system, which degrades the quality of his spoofing and increases detection performance. Even if the spoofer accounts for the existence of white noise in his data, the algorithm is highly sensible to the noise and detection is increased significantly.

This results show some scenarios with possible application of the algorithm to detect spoofing faults with several variations. This algorithm however will have it's limitations and the performance is limited by the IMU precision and grade. This first practical implementation used a Low tactical grade IMU and if upgraded, detection would increase.

The first implementation of the algorithm will set the base for further investigation on the variations of the algorithm to different scenarios so the detection quality can be assessed and characterized not only through theory but through experimentation.

6.1 Future Work

Even after this experiments, the work could be expanded to a more extensive report against spoofing testing the performance of the algorithm to variations

in the worst case fault instead of a generic fault.

The same variations that we studied could be applied to the worst case fault to study the benign and malignant effects of the spoofer or vehicle lack of perfect knowledge.

Several improvements in the code could be added and studied. For example the addition of atmospheric delays like Tropospheric and Ionospheric delays and it's effects on detection. This additions would provide more insight on the behaviour of the estimator in a more realistic environment if the models for the delays are not exactly known.

Later the addition of multipath perturbations could be used to test environments in crowded urban canyons or airborne applications, each with particular multipath behaviours and parameters.

Lastly, combining all the effects mentioned would result in a highly in depth study of the innovation sequence based INS estimator and the performance in real environments live could be implemented and tested.

This however laid outside the scope of this research and it is encouraged for future researchers to pursue these studies to gain further insight in the applicability of this technologies in the navigation environment.

References

[1] J. A. Farrell, *Aided Navigation.* Mc Graw Hill, 2008.

[2] M. Salvemini, "Global Positioning System," in *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*, vol. 10, pp. 174–177, Elsevier, 2015.

[3] R. G. Brown and P. T. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, Inc., 1985.

[4] Ç. Tanil, "Detecting GNSS Spoofing Attacks Using INS Coupling," *Thesis*, 2016.